NAIST-IS-DD1461215

Doctoral Dissertation

Human Action Recognition-Based Summarization of User-Generated Sports Video

Antonio Tejero-de-Pablos

March 15, 2017

Department of Information Science
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Antonio Tejero-de-Pablos

Thesis Committee:

Professor Naokazu Yokoya	(Supervisor)
Professor Hirokazu Kato	(Co-supervisor)
Associate Professor Tomokazu Sato	(Co-supervisor)
Affiliate Assistant Professor Yuta Nakashima	(Co-supervisor)

Human Action Recognition-Based Summarization of User-Generated Sports Video*

Antonio Tejero-de-Pablos

Abstract

Nowadays, a vast amount of videos are taken due to the exponential growth of commercial devices capable of video recording. The main targets of such videos include sports that users may record in, e.g., a public event or a professional match. These videos are usually long, containing redundant and uninteresting parts, and thus they are often stored and never reviewed again. The field of sports video summarization allows to automatically extract the highlights of the original video for a quick review. Existing work in this field leverages various knowledge in application domains, e.g., structure of games and editing conventions, which are commonly found in broadcast video. However, users' self-recorded videos normally lack any kind of editing conventions and the structure of the sport is sometimes lost, and thus the existing work is ineffective.

This thesis approaches the challenge of summarizing self-recorded sports video by resorting to the field of human action recognition (HAR). We hypothesize that players' actions can be recognized and used as a novel source of semantics to elaborate summaries. The greatest difficulty in HAR is to deal with the variability in the actors' anthropometry and the camera viewpoint. This can be alleviated by using depth information, obtainable by widely used commodity RGB-D sensors (e.g., MS Kinect). The state-of-the-art works in HAR use classifiers that require a large amount of training data, but in some cases we may not have such a big dataset, e.g., when using a self-recorded one.

*Doctoral Dissertation, Graduate School of Information Science,
Nara Institute of Science and Technology, NAIST-IS-DD1461215, March 15, 2017.

In this thesis, we first propose an HAR method with flexible learning that does not require a large number of training instances to perform recognition. Unlike other methods, ours successfully deals with the trade-off between accuracy and flexibility. Then, we propose a novel user-generated sports video summarization method that acquires higher level semantics of the video by applying the HAR to RGB-D video sequences. We use the recognition results of the players' actions to model the interestingness of the lengthy original sequence and extract the highlights of the game. We deal with the limited number of instances of our self-recorded HAR dataset by using the aforementioned flexible HAR method. We target sports that consist of a series of actions, such as tennis, boxing, and martial arts. We took Kendo as an example sport to evaluate our method, and investigated the accuracy and quality of the generated summaries objectively and subjectively. We trained our novel highlights extraction model with the subjective opinion of groups of users with different experience in the sport, and studied the adequacy of our method to each group. We also studied the effect of employing RGB and depth information together and separately when modeling interestingness through the use of deep learning.

Keywords:

video summarization, human action recognition, sports video, user-generated video, RGB-D camera

Contents

List of Figures

List of Tables

Glossary

2D – two dimensional

3D – three dimensional

ART – adaptive resonance theory

AT – action template

CMU – Carnegie Mellon University

CNN – convolutional neural network

DTW – dynamic time warping

FN – false negative

FP – false positive

GMM – gaussian mixture model

GPU – graphics processing unit

GPGPU – general-purpose computing on GPU

HAR – human action recognition

HOG – histogram of oriented gradients

HMM – hidden Markov model

ISA – independent subspace analysis

LCSS – longest common subsequence

LOO – leave one out

LSTM – long short-term memory

MMTW – maximum margin temporal warping

MoCap – motion capture

MS – Microsoft

MSR – Microsoft research

NN – nearest neighbors

RGB – red green blue

RGB-D – red green blue depth

RNN – recurrent neural network

SIFT – scale-invariant feature transform

SVM – support vector machine

TN – true negative

TP – true positive

UGSV – user generated sports video

UGV – user generated video

1 Introduction

We live in an era where cameras and commercial devices capable of video recording are widespread available in a variety of sizes and functionalities. This has lead to an ever-growing enormous collection of unedited and unstructured video data generated by users around the world [40,100]. Among them, sports video appeals to large audiences, being one of the most popular themes. Nowadays users can take sports video with their own devices at public events, professional matches, etc. These user-generated videos are normally lengthy, with a lot of redundant and uninteresting parts, and therefore they require summarization for an easier review. Also, by reducing their size, we facilitate the distribution of the video through different online platforms (e.g. social networks). Nevertheless, manually extracting video highlights, i.e., the most interesting contents of the video, is a very time-consuming task. In order to tackle this problem, the field of automatic video summarization [102] studies techniques to automatically compact the content of a video to facilitate its storage, transmission, browsing, etc. Researchers have studied sports video summarization for decades, and they have proposed several methods for creating a summary with the interesting highlights of a sports game [28, 47, 72]. Most of these methods are specific for broadcast video, since it is edited following sport-specific conventions that are easily detected and can be used to find the highlights of the game. For example, television programs, which are recorded and edited by an expert, feature slow-motion replays, narration, superimposed text, and fixed camera angles that imply a free kick in soccer or a pitch in baseball [18]. Also, some sports like baseball and American football have a certain structure in a game itself [47, 72], which can be also used to extract the moments of greatest interest in a game, and create a summary. For example, in baseball, pitching and batting scenes intertwine in a way that is common to all broadcasts.

However, in contrast to broadcast video, normally user-generated sports video does not follow any convention, and the structure of the sport is not always well defined. The computer vision community has proposed several approaches to understand the content of unstructured video and user-generated video (UGV). These approaches range from the traditional clustering of video features that eliminates redundancy, to the most recent works that use deep neural networks to automatically learn features that allow modeling the interesting segments of the video [49,116]. However, to the best of our knowledge the problem of summarizing user-generated sports video has not been directly tackled to date. In order to approach the problem of summarizing UGV of sports, we need to rely in a source of features that should not depend on any editing convention and yet should be present in every sports video. We, as a novel approach for video summarization, propose to use the *players* as our source of features, more concretely, their *actions*.

The area of computer vision that studies how to model and classify human actions from video is called human action recognition. Human action recognition (HAR) attracts the attention of many researchers due to its numerous applications, such as video surveillance and human computer interaction [103]. However, providing a machine the ability to recognize human actions from an image sequence is a challenging task due to their large variability in various factors [1]. In [88], the authors identify three main sources of variability in human actions: viewpoint, execution rate/speed, and anthropometry.

While HAR has been traditionally applied to color images [8], the recent commodification of depth sensors provides a way to reduce the variability using depth information [48, 77]. They provide 3D structure of scenes, which facilitates the understanding of human actions under conditions in which 2D approaches may be ineffective (e.g. motion perpendicular to the camera plane). Moreover, depth sensors have opened a door for the development of novel techniques that have been used in many computer vision-related research [31, 95]. A distinguished technique, especially advantageous for HAR, is 3D articulated skeleton tracking in real-time such as [89], which allows modeling human actions in terms of trajectories of body joints. This method is more reliable than using other visual features that are tied to the user's appearance, such as silhouettes. Various techniques have been proposed using depth sensors [107, 113], and more specifically,

human joint models. They use different types of classifiers such as hidden Markov models (HMMs) and support vector machines (SVMs). Most of these recognition methods rely on an expensive learning process with a large training dataset for generalization performance. However, some applications may not count with a large number of instances to be trained with or may need flexibility in learning and classifying the user's behavior (i.e. learning of new actions during runtime).

In this thesis, we hypothesize that using human action recognition techniques we can obtain a representation of the players' actions in a video by which we can model the interesting highlights. For example, a boxing scene showing a parry and an aggressive uppercut might be more interesting than a scene showing a feint or a failed attack. With this idea in mind, we propose a first methodology for which we recorded our own UGV of sports using a commercial RGB-D camera. The 3D information provided us with accurate information on the movements of players, but the dataset was not big enough to train current action classifiers. We then came across with challenge of designing a flexible action recognition method that could provide state of the art accuracy without requiring too many training instances, as we mentioned above.

Once we overcame this challenge and proved our theory, another issue remained. Although we believe that in the near future smartphones and other everyday devices will be equipped with technology able to capture three dimensional information, currently most UGV contain only color images (2D). We were pushed to explore new methods for extracting players' actions from UGV so that they allow us to model highlights. Motivated by the outstanding results of convolutional and recurrent neural networks, the latest fashion in image and video processing, we propose another approach for user-generated sports video summarization in order to surpass our previous method.

The remainder of this thesis is organized as follows. First, Chap. 2 reviews the state of the art in sports video summarization, user-generated video summarization and human action recognition. Then, Chap. 3 presents our approach for flexible HAR using estimated 3D body joint positions that deals with the trade-off between flexibility and accuracy. In Chap. 4 we use the HAR method presented in Chap. 3 and present a novel approach for video summarization that aims to recognize players' actions to model the highlights of a sports game. Continuing the

work in the previous chapter, Chap. 5 describes an improved methodology for motion feature extraction and highlights modeling for summarizing user-generated sports video. Finally, Chap. 6 draws the main lessons learned from this thesis and outlines several future work.

2 Related work

The computer vision community has studied video processing tasks for a long time. Tasks such as action recognition [111], abnormal detection [90], activity recognition [36] and video summarization [14] have a point in common, the problem of feature representation of video. This section reviews the main state-of-the-art works in the fields of sports video summarization, UGV summarization and human action recognition, and states the contributions of our method.

2.1 User-generated sports video summarization

2.1.1 Sports video summarization

Summarization of sports video focuses on extracting the most interesting moments, or highlights, of a sports game/match. One of the major approaches to analyze sports video for summarization is using editing conventions present in broadcast programs, which are common to almost all videos of a specific sport [18]. In [28], the authors proposed summarizing broadcast soccer video based on editing conventions and detection of soccer field elements (e.g., goal) (Figure 2.1). Leveraging editing conventions allows finding the highlights of a sports game easily [16,98]. For example, a slow-motion replay may indicate the presence of a key point in the game [74], or certain pre-defined camera motion patterns can indicate a shot in basketball and soccer [115] (Figure 2.2). Other methods use the "play" structure of certain sports. In [47], authors use the "play" structure of American football, baseball and sumo wrestling for modeling their video highlights. These "plays" are defined according to the rules of the sport (i.e., a touchdown in american football), and can be detected based on the conventional patterns of broadcast video (Figure 2.3). Other works leverage the metadata of sports videos

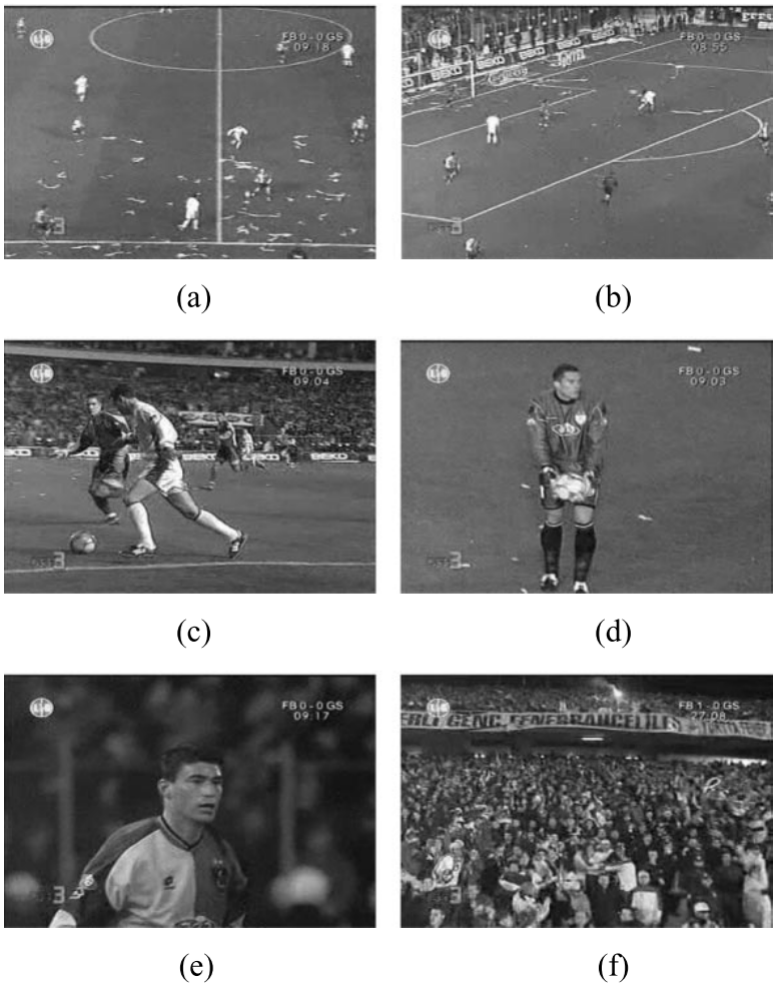


Figure 2.1: Viewtypes in soccer: (a), (b) Long view, (c), (d) in-field medium view, (e) close-up view, and (f) out of field view. Obtained from [28].

that use the MPEG-7 codec [26, 72], since it contains play information such as the inning structure in a baseball game (Figure 2.4).

All the aforementioned summarization methods are based on domain dependent heuristics, which makes them hard, if not impossible, to generalize to other sports. This type of approaches represent the majority of sports video summarization methods to date. However, they cannot be applied to UGV due to their lack of structure and other conventions. Just very few methods have used motion as a non-heuristic feature to generate the summary of a sports video. [60] uses a very

Figure 2.2: Key frame of clips for (a) horizontal motion, (b) vertical motion (c) radial motion. Obtained from [115].

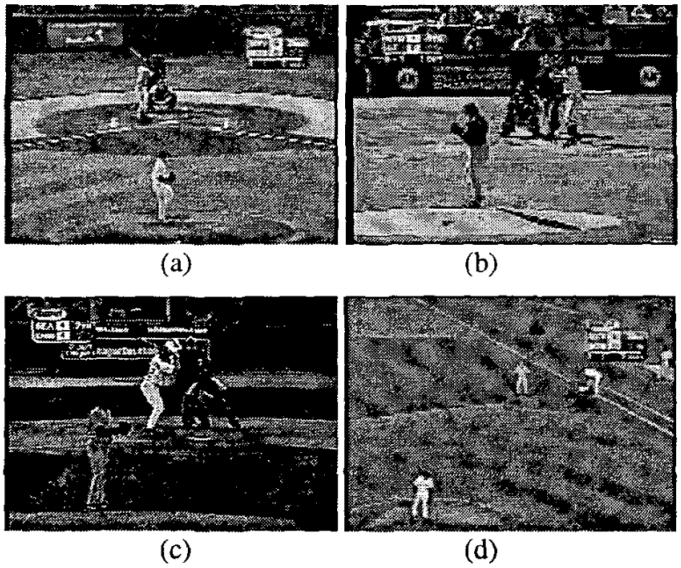


Figure 2.3: (a)-(c) A typical start of a regular play - a pitching scene. Other types of starts include a base-stealing scene (d), which is also captured from a fixed camera angle. Obtained from [47].

simple approach, taking the local minima of an optical flow function for keyframe extraction in rugby videos. In a similar fashion, [12] calculates the direction of the variations of the activity level in the color frames to represent how lively the scene changes [25, 38, 44, 67], and then segment semantically relevant events in broadcast games of soccer, basketball, and tennis. The results are acceptable but do not allow to capture the most interesting highlights of a sports game. In an attempt of performing a more precise semantic analysis of sports game, [121]

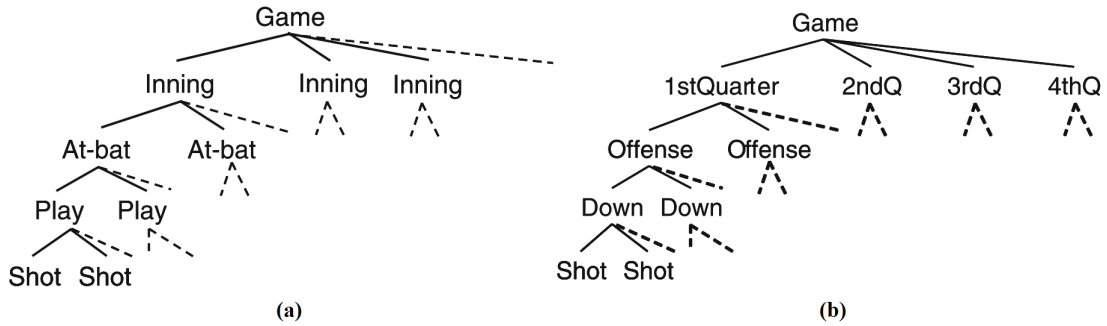


Figure 2.4: Game tree of a (a) baseball video and (b) an American football video. Obtained from [72].

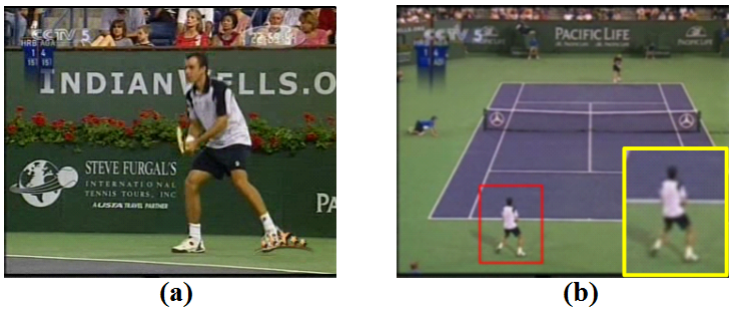


Figure 2.5: Two typical frames derived from broadcast tennis video. (a) Close-up, (b) Far-view, the zoomed picture is the player whose action is recognized. Obtained from [121].

used action recognition on tennis players' actions in combination with editing conventions (Figure 2.5). However, due to the difficulty of recognizing actions from the RGB video frames they were able to recognize only two action classes, left swing and right swing.

In the next section we explain which methodologies are currently applied to summarize UGV, which needs to be approached in a different way.

2.1.2 User-generated video summarization

Unlike conventional sports video, which is normally edited according to the conventions of the sport (i.e. fixed camera angles, replays, etc.), user-generated video (UGV) does not necessarily follow any particular convention, structure or image

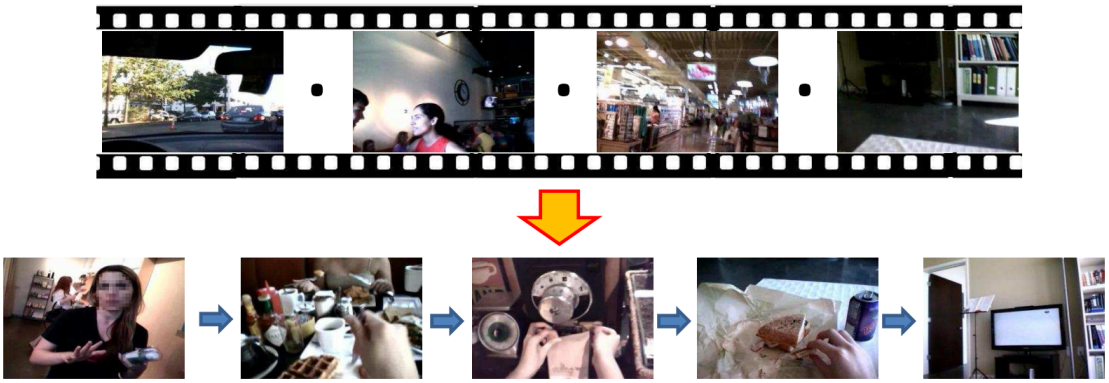


Figure 2.6: From an input egocentric video, a storyboard summary of important people and objects is calculated. Obtained from [46].

quality standards [38]. Summarization methods cannot rely on audio either, since it is usually noisy. For this reason, instead of trying to understand the contents, traditionally UGV summarization methods avoid any semantic interpretation of the video with methods such as uniform frame sampling [62] and video feature clustering [49]. This way, most of the work in UGV summarization try to convey a brief but representative synopsis of a lengthy video, given priority to diversity of the extracted segments. However, these segments do not always correspond to the highlights, which depend heavily on the video domain.

Most highlight detection works focus on broadcasting sports video [35, 68, 83, 97, 109, 114, 117], but as we said in the previous section, we cannot use the same techniques when dealing with UGV. For this reason, other types of features for highlights modeling have been explored since then, such as interestingness [34,70], important objects [46] (Figure 2.6) and attention [55]. Related to the latter, some works model video highlights based on the viewers' preferences, which can be obtained explicitly from viewers or inferred from their reactions while watching the video [76]. Recent work in [116] approaches highlight detection in UGV by learning which video segments contain semantics that are more interesting to the user discarding those who are not (Figure 2.7). The authors try to avoid heuristic rules (e.g. detecting the presence of the bride in a wedding video [17]) since they do not generalize well to generic, unstructured videos. Instead, they leverage of deep neural networks for unsupervised summarization. Recent advances in re-

Figure 2.7: Highlight detection results in different video domains using RNN. The red borders indicate snippets detected as highlights. Obtained from [116].

current neural networks (RNN) allowed more sophisticated UGV captioning and summarization approaches, including the generation of video titles and descriptions. In [93] they automatically generate a highlight summary using the title of the video as a reference, and viceversa in [118]. These works use LSTM [37], an RNN that effectively models temporal relationships of the extracted features.

Video feature extraction have also been benefited from deep learning and, more specifically, convolutional neural networks (CNN). The trend in recent UGV summarization works is to leverage learned features obtained with a CNN that is pre-trained with a large dataset. In [118], they use the well-known VGG network [92] pre-trained with the also popular Imagenet [39] in order to extract features of user generated videos (Figure 2.8). The problem with this network is that it lacks any kind of motion modeling, focusing only in the appearance of the video frames. However, in a genre such as sports video motion plays a very important role, so motion and human actions should be also considered when extracting features to model video highlights.

User-generated sports video summarization presents a new challenge that needs to be approached taking into account the intricacies of summarizing sports video and UGV. In the absence of any heuristics, user-generated sports video summarization has been approached by extracting semantics from a different source, i.e. the players' actions. In [23], highlights are modeled by applying human action recognition to the players' actions and learning the sequences of moves and techniques that the user considers interesting. Although some current methods for highlight extraction of UGV use features based in motion (e.g., [116]), it is un-

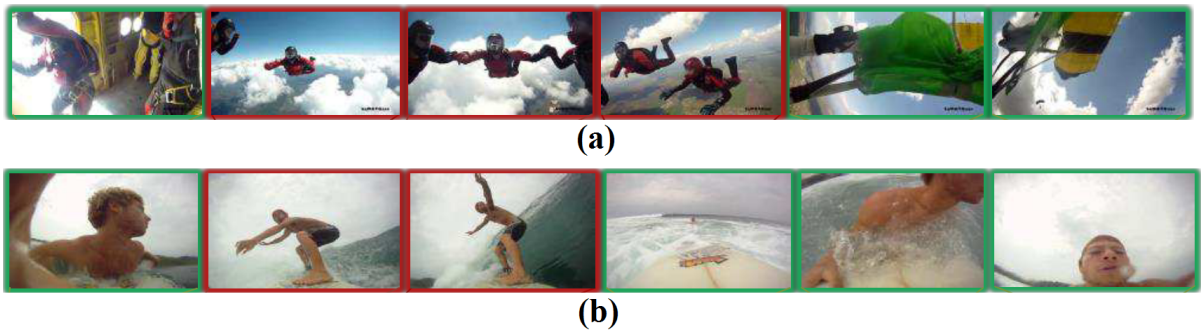


Figure 2.8: The highlights detection results are clearly based on the general appearance of the scene. Obtained from [118].

likely that these methods can differentiate between actions at a higher level (e.g. different sequences of punches and defenses of both players in boxing). To the best of our knowledge, ours [23] is the only work that approached user-generated sports video summarization directly.

Next section reviews the state-of-the-art works in human action recognition, and how we can make use of them to summarize UGV of sports.

2.2 Human action recognition

2.2.1 HAR in RGB video

Until recently, HAR has been performed exclusively on videos captured with traditional cameras [8]. Some methods directly use captured images as spatio-temporal volumes to represent motion [13, 61, 86]. In [9], Calderara et al. extracted trajectories from 2D images to represent how human motion varies in time (Figure 2.9). However, finer action recognition requires to segment the human body and extract the pose information. Once the body model is obtained from the video, different features related to the human pose can be extracted. Fujiyoshi et al. [30] and Chen et al. [15] extract a primitive skeleton for modeling human actions, in which the skeleton is simplified for reducing the computational cost (Figure 2.10).

However, these methods suffer from some inaccuracies in the processing of RGB images. According to [71], estimating human poses from 2D video is harsh due to

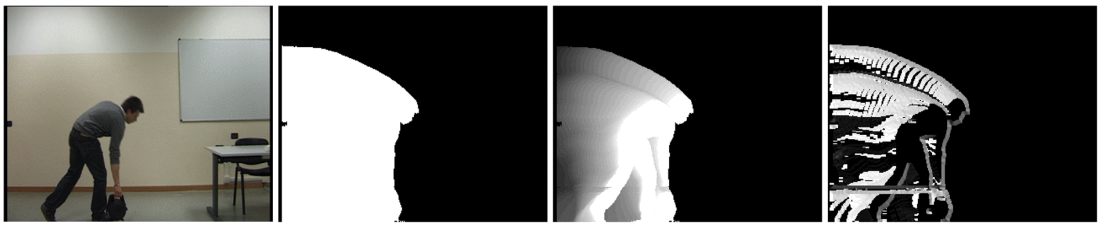


Figure 2.9: 2D representation of human motion. Obtained from [9].

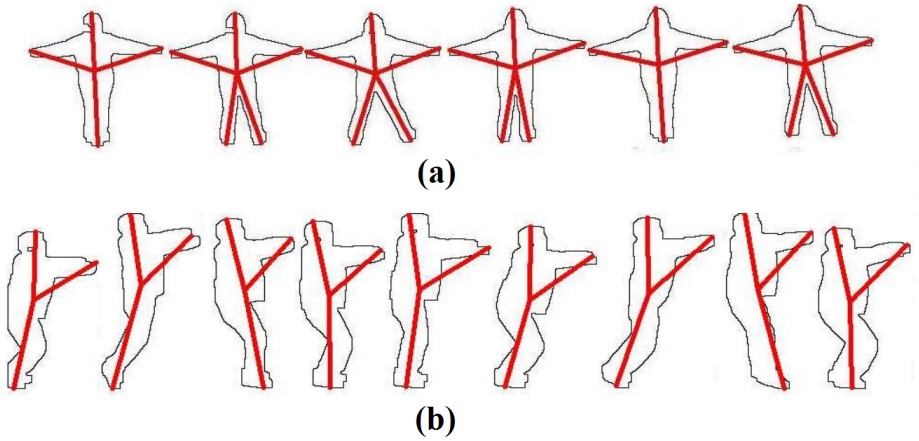


Figure 2.10: Simple skeleton representation obtained from 2D images for actions: (a) Sidewalk and (b) Jump actions. Obtained from [15].

large variations in appearance. In addition, the segmentation of human figures in order to estimate the pose in RGB images is very computationally expensive, due to the high dimensionality of visual features [122]. In the same manner, since the estimation of explicit positions of body parts in a continuous way is difficult, it is also hard to create a general algorithm to learn the model parameters of human actions. It should be noted that the main limitation of such 2D methods is that poses are captured from a single point of view [81], and therefore, certain types of actions can be highly ambiguous.

2.2.2 HAR in depth video

With the release of commodity depth sensors, HAR underwent a breakthrough thanks to the application of additional 3D information [2]. The use of depth maps

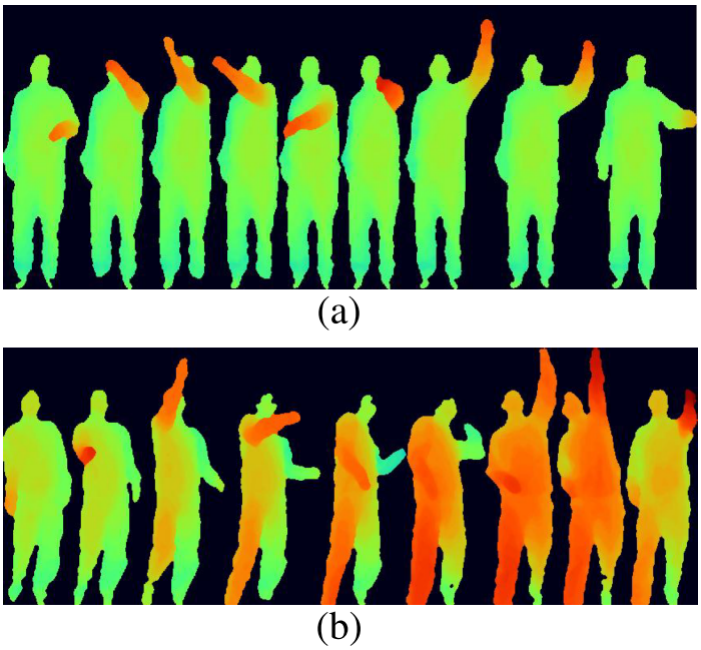


Figure 2.11: Examples of the sequences of depth maps for actions: (a) Draw tick and (b) Tennis serve. Obtained from [48].

alleviates variations in human appearance to a great extent [7]; they can make human segmentation in video far easier and almost immune to illumination, camera blurring, and other factors that hinder HAR. Based on these premises, Li et al. [48] used a depth sensor to obtain a depth map sequence, which is represented as a bag-of-3D-points in order to model the actions (Figure 2.11). Although it outperformed 2D methods, including other bag-of-words-based representations such as [27], this method is still view-dependent because the sampling is performed directly on the depth maps. Another technique involves applying histograms to the 3D point cloud sequences captured by the sensor to calculate descriptors that characterize human shape motion, such as histograms of 4D normals [73] and principal components [80].

One of the advantages of using depth sensors for HAR is that it facilitates the estimation of accurate 3D body joint positions from depth maps via skeleton tracking (Figure 2.12). These 3D positions can be more direct cues for HAR, providing robustness against variations in viewpoints. Such 3D body joint trajectories used to be available only with expensive equipments such as motion

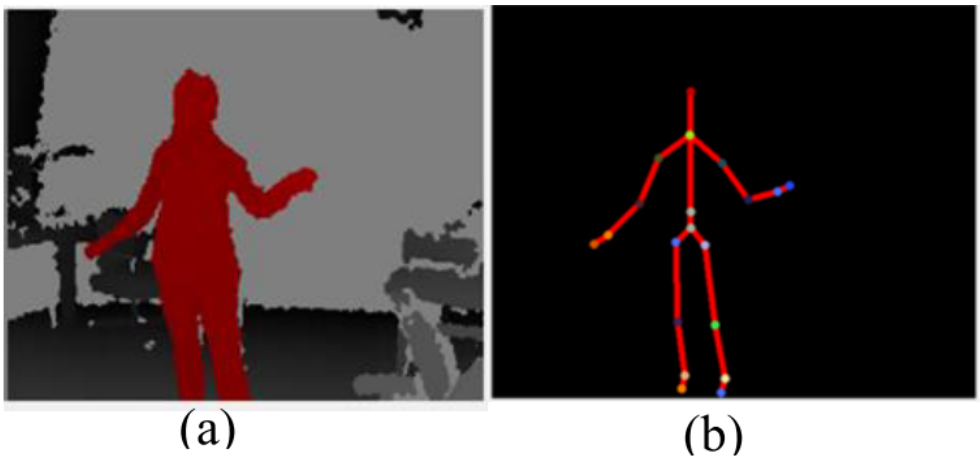


Figure 2.12: (a) A depth image and (b) its corresponding estimated 3D body joint positions. Obtained from [113].

capture devices (MoCap) [19], as in [54]. But currently they are obtainable with commodity RGB-D sensors with built-in real-time 3D human tracking capabilities (e.g. Microsoft Kinect), although the tracking is not exempt from errors [89]. For example, Xia et al. [113] proposed to use the body joints provided by Kinect to perform HAR using HMMs.

Martínez-Zarzuela et al. [59] and Wang et al. [107] use the discrete Fourier transform to represent the joint trajectories in the frequency domain and then feed them into a classifier, Fuzzy ARTMAP [10] and support vector machines (SVMs) [96] respectively. The discrete Fourier transform reduces the dimensionality of the joint trajectories by assuming that the most crucial information is concentrated in the lower frequency components. It also reduces noises due to tracking errors, which is a problem inherent to joint estimation from depth maps.

Variations in execution rate of human actions have a negative impact in HAR [104]. Many works have relied on DTW to gain robustness against these variations. Müller and Röder [66] used DTW to build semantically interpretable action models by extracting relational features that encode temporal dynamics. These relational features (e.g. the right hand is up or down) exclude a lot of detail of the action, but retain view-invariant information about the overall configuration of a pose for its classification. However, because of the loss of detail, this method confuses actions when they are too similar or too short, and the accuracy is very

dependent on the manually designed features. In [108], Wang and Wu dealt with variations in execution rate by combining an SVM-based classification algorithm with DTW. Alternatively to DTW, longest common subsequence (LCSS) is used in [75] to make their action classifier invariant to temporal variations. In [4], the authors find a representation of the body joint trajectories that is robust against execution rate variations among subjects. They consider two HAR schemes, a nearest neighbors (NN) classifier and an SVM classifier.

2.2.3 Flexible HAR applications

There are a range of HAR-based applications that require learning new actions in runtime. Applications such as customizable gesture interfaces [52, 63] and action databases, either for indexing or retrieval [65,66], can benefit from such capability, since they are expected to be able to recognize a new type of action right after being input. This kind of applications also does not count with many learning instances [78]. Hence in this thesis we consider the flexibility of approach by two factors:

- Being able to learn a certain action class at runtime.
- Being able to recognize actions even with a very small number of training instances.

We consider that a method is capable of runtime learning if it does not perform any optimization of the classifier when learning a new action instance. The majority of the previously mentioned works rely on classifiers with a costly learning process that cannot be updated at runtime (e.g. SVM) and therefore are not suitable for applications that require adaptive modification of the training model. On the other hand, methods that are capable of runtime learning (e.g. NN) allow this, but to the best of our knowledge they have not proved state-of-the-art accuracy yet.

2.3 Action feature extracion from UGV of sports using deep learning

To summarize the work exposed in Section 2.2, traditional works in human action recognition (HAR) used hand-crafted features from color frames to model human movement from RGB video [13, 87]. With the commodification of depth cameras (e.g., MS Kinect), the trend in HAR became leveraging three-dimensional information in order to disambiguate movements parallel to the camera plane [5] and gain robustness to occlusions, variability in lighting conditions, etc. [7]. These works use either depth maps [48, 59, 73] or 3D joints estimated from the depth maps [107, 113] to obtain the highest accuracy in action datasets such as MSRAction3D and MSRDailyActivity3D.

However, with the exponential grow of UGV, the trend of deep learning also influenced the field of HAR. The advantage of using CNN over methods that use hand-designed local features, such as SIFT [53, 84], HOG [22, 43] or dense trajectories [105], is that CNN learn directly from data and consequently the extracted features are more generalizable to many domains. Contrary to what one might think, this does not make CNN less accurate than hand-crafted features; the state of the art shows how CNN-based methods trained using large-scale action datasets are able to outperform them for several types of tasks (e.g., object detection, action recognition, etc.) [101].

It is also known that 3D CNN are more suitable for spatiotemporal feature learning compared to 2D CNN [101]. Whereas the CNN used to extract features from images is two-dimensional, a three-dimensional CNN is an extension that includes the temporal dimension, and it is used in the case of video. In [45], a 3D convolutional neural network along with used independent subspace analysis (CNN-ISA) is used to recognize human actions from realistic video datasets. Also in [101], authors introduce a CNN called C3D that they use to extract features from action videos. C3D is generic on capturing appearance and motion information from videos and it can be used for different tasks depending on the dataset it is trained with (i.e., 1Msports for action recognition, YUPENN for scene recognition).

The latest trend in CNN-based HAR methods is utilizing two types of streams,

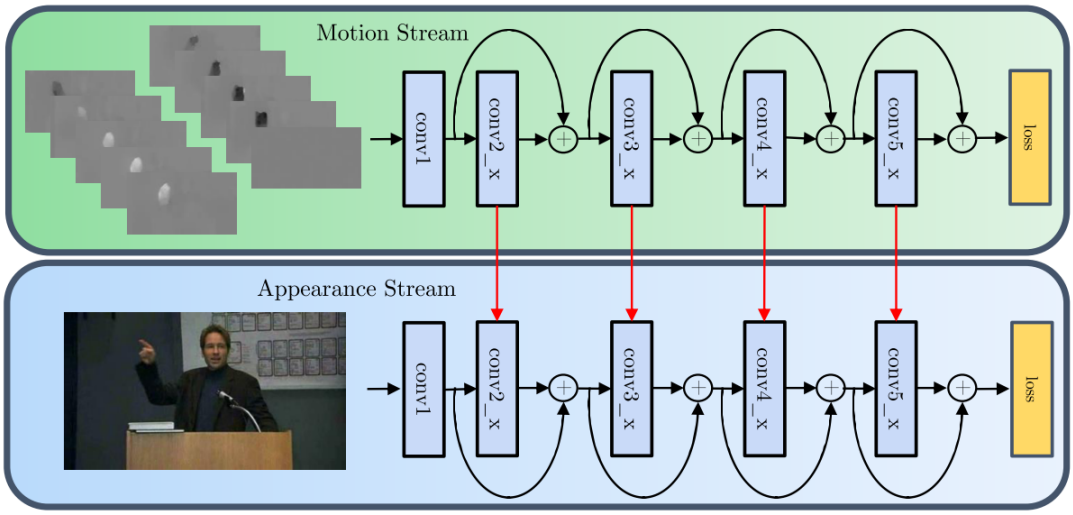


Figure 2.13: Example of a two-stream CNN that separately captures appearance and motion. Obtained from [29].

a spacial *appearance* stream for representation of images, and a temporal *motion* stream for representation of local motion features [29, 91] (Figure 2.13). First, video is decomposed into spatial and temporal components by using RGB and optical flow frames. These are fed into separate CNNs to learn spatial and temporal information of the objects in the scene. Each stream performs video recognition separately and softmax scores are combined for classification. This architecture is supported by the two-stream hypothesis of neuroscience, in which the human visual system would be composed of two different streams in the brain, the dorsal stream (spatial awareness and guidance of actions) and the ventral stream (object recognition and form representation) [32] (Figure 2.14).

2.4 Contributions of this thesis

In this thesis, we propose and evaluate a novel HAR approach focused on flexibility. Our method is based on the nearest neighbor (NN) approach [21] and uses the joint trajectories estimated from the depth maps, referred to as action templates (ATs), as a model for each action class. Unlike other state-of-the-art methods, ours does not require a computationally expensive learning process; modification

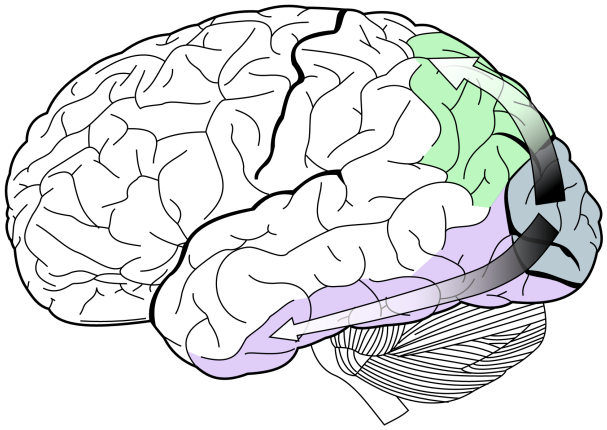


Figure 2.14: The human visual system is composed of the dorsal stream (green) and the ventral stream (purple). The dorsal stream is responsible for detection of location and motion. Obtained from *Wikipedia: Motion perception*.

of the model can be done by just adding new labeled joint trajectories to the set of ATs. For action classification of an unknown action sequence, our method calculates the distance between that sequence and each AT via dynamic time warping (DTW), which is widely used for analyzing time series data [79]. The joint trajectories estimated from depth maps are generally noisy, which might hinder recognition accuracy. For this reason, we include in our ATs the confidence values of each tracked joint along with their respective position, and modify the DTW algorithm to calculate a distance between actions while avoiding erroneous trajectory sections.

We also propose a novel method for user-generated sports video summarization using a new source of semantics extraction, *i.e.*, depth of scenes, which becomes available and affordable due to the recent development of RGB-D sensors including Microsoft (MS) Kinect. More specifically, some sports, such as tennis, boxing, and martial arts, consist of a series of actions (*e.g.*, uppercut, and jump-kick), and our method automatically labels them by applying human action recognition (HAR) to RGB-D video sequences. Unlike other state-of-the-art methods, ours does not require We model the highlights of a game based on HAR results to extract them from a lengthy original RGB-D video. To the best of our knowledge,

this is the first attempt to use this kind of analysis for video summarization. We evaluate our method objectively and subjectively, surveying users with different experience in the sport.

Finally, we describe an improved method for user-generated sports video summarization. Inspired in the latest trends in deep learning for HAR, we propose a two-stream architecture that uses detailed and coarse motion features. We study a range of representations of actions from video, from human pose estimation using depth maps to learning spatiotemporal features for videos using convolutional networks trained on large-scale RGB video datasets. We model the highlights of our videos by learning the temporal relationship of our features through a recurrent neural network designed for this particular task. We surveyed users with different levels of experience in the sport to investigate the adequacy of our method to their particular preferences, and compared it to our previous approach. Table 2.1 shows the comparison of our method with other sports video summarization methods according to their requirements.

Table 2.1: Comparison of sports video summarization methods.

Method	Structure of the sport	Post-editing	Predefined camera angles	Player action recognition
MPEG-7 metadata [72]	Yes	Yes	No	No
"Play" detection [47]	Yes	No	No	No
Shot detection [115]	No	No	Yes	No
Narration/text [28]	No	Yes	No	No
Optical flow variations [60]	No	No	Yes	No
Our method [23]	**No**	**No**	**No**	**Yes**

3 HAR for RGB-D video datasets with a reduced number of instances

3.1 Overview

As we introduced in Chap. 1, our first approach to sports video summarization should be able to recognize players' actions to model video highlights. However, due to the lack of a user-generated benchmark of sports video, we decided to create our own annotated dataset. As many other self-recorded datasets, ours did not have a number of instances big enough to train a very sophisticated method, although we still needed to recognize actions. This motivated the work presented in this chapter. We introduce a novel action recognition aproach that uses 3D joints estimated from depth maps in RGB-D video. The novelty of our method lies in its flexibility to learn new instances and its capability of recognizing actions even with a reduced number of learned instances. Several methods can take advantage of these benefits, such as applications that need to learn new actions in real-time, or like in our case, applications with a reduced number of training instances. Besides, the use of 3D positions assures more accuracy when recognizing actions, especially those perpendicular to the camera plane.

3.2 Flexible HAR using masked 3D joint trajectories

Fig. 3.1 depicts an overview of our method, which takes a nearest neighbor-based approach to gain flexibility instead of learning a classifier for each action class. We first estimate the 3D joint positions using skeleton tracking from a series of depth map sequences using, e.g., [120], and store them with their action labels as instances of a training dataset. One of the main issues that lead to failure in HAR is concerned with the estimation errors in the skeleton tracking, as stated in [107]. Fortunately, the joint position estimation algorithm provides a confidence value for each joint tracked in each frame. Our method uses it for both learning and recognition stages to alleviate the problem of erroneous skeleton tracking. Then, we prepare an AT for each given action class, which can be viewed as a model of a specific action. Each AT consists of a set of joint trajectories of the action instances belonging to that class along with the confidence values for each joint positions.

At the recognition stage our method tracks the joint trajectories of an unknown action instance in the same way as the learning process, and retrieves its closest instance from the ATs in the database. Since different instances of the same action can be subjected to temporal variations (especially different length and execution speed), we employ a DTW-based distance measure for template matching during the nearest neighbor-based classification.

3.2.1 Action templates learning

To generate an AT, we manually select $J = 15$ different joints from the skeleton tracked in an action instance, as illustrated in Fig. 3.2. Let $\mathbf{p}'_{fj} = (x_{fj}, y_{fj}, z_{fj})^\top$ denote the 3D position of joint j at frame f. Since these positions are in the RGB-D sensor's coordinate system, they can vary from one action instance to another depending on the position of the actor relative to the sensor. For reducing this variability, we transform the joint coordinates so that a certain joint coincides with the origin to improve the robustness against viewpoint variations. In this work we choose the torso as the origin, thus denoting the transformed joint position as $\mathbf{p}_{fj} = \mathbf{p}'_{fj} - \mathbf{p}'_{f\text{torso}}$.

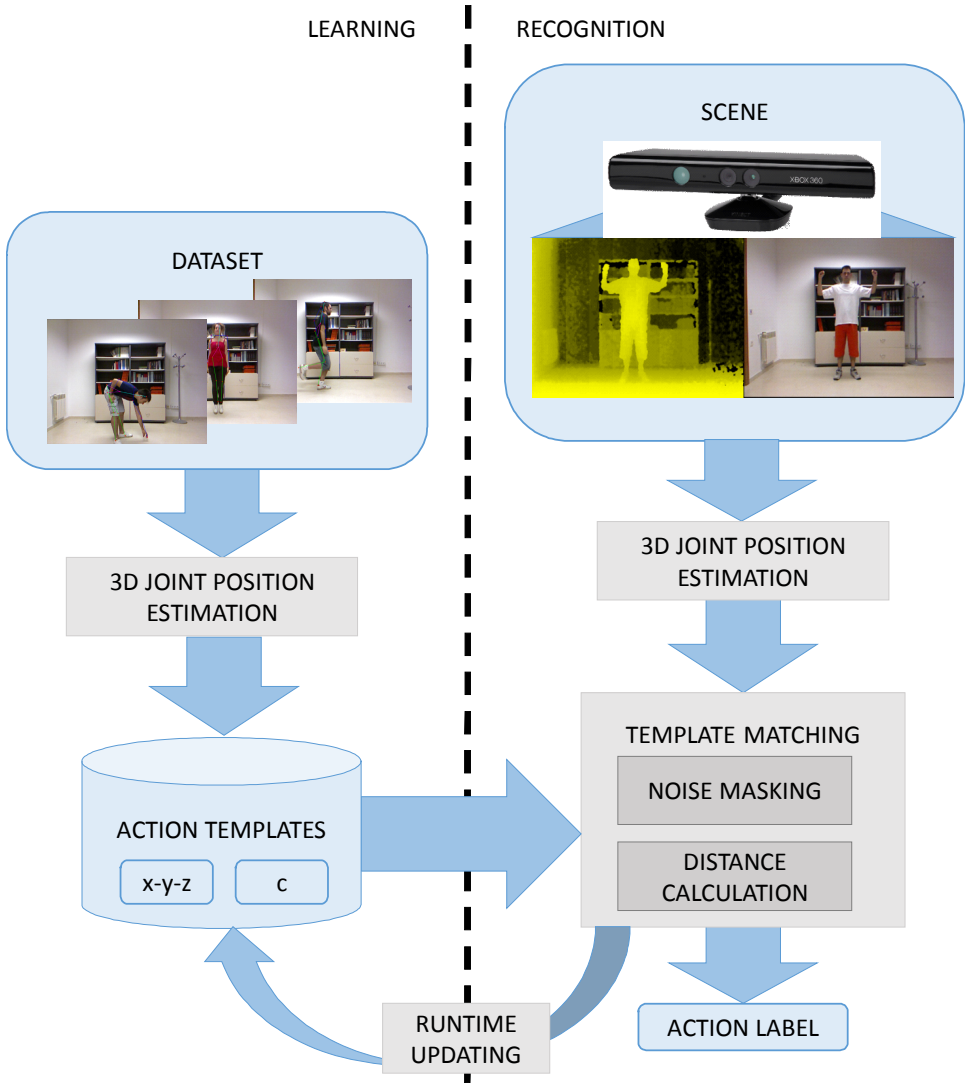
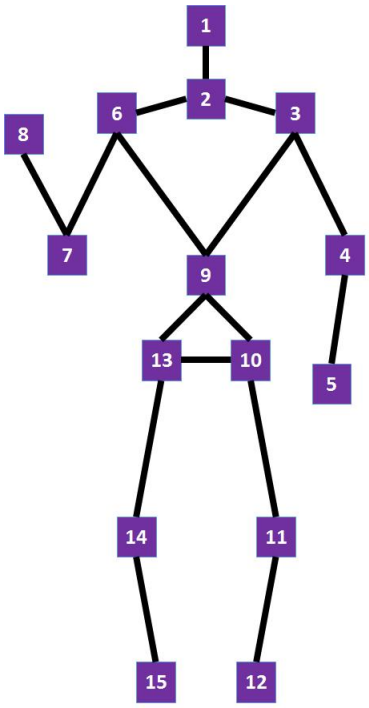


Figure 3.1: Overview of our HAR method. The 3D joint positions (x,y,z) along with the confidence value (c) are tracked from the video source to build action templates for each action class. They are used to match new actions and updated at runtime.

The joint trajectories of all the instances from a certain action class are then aggregated to form an AT. Along with them, the associated confidence values of the tracked positions offered by the joint estimation algorithm of the skeleton tracker [120] are also included. Let m_i be the action class label for the joint



1	Head
2	Neck
3	Left shoulder
4	Left elbow
5	Left hand
6	Right shoulder
7	Right elbow
8	Right hand
9	Torso
10	Left hip
11	Left knee
12	Left foot
13	Right hip
14	Right knee
15	Right foot

Figure 3.2: Human body joints used in our method. Numbering and names correspond to the ones used in the skeleton tracker.

trajectories of the instance i in the training dataset ($m_i = running$, for example), $P_i = \{\mathbf{p}^i_{fj} | f = 1, \ldots, F_i, j = 1, \ldots, J\}$ the corresponding joint trajectories, and $C_i = \{\mathbf{c}^i_{fj} | f = 1, \ldots, F_i, j = 1, \ldots, J\}$ their corresponding confidences, where F_i is the number of frames for action instance i. The AT for action class M is then a set of joint trajectories with their respective confidence values, i.e.,

$$A_M = \{(P_i, C_i) | i \text{ s.t. } m_i = M\}. \tag{3.1}$$

The learning process only requires the generation of ATs.

3.2.2 Action classification

Our recognition process calculates a distance measure to find in our ATs the action instance that is the nearest neighbor of the given unknown instance. Due to the variability in the execution of human actions, naive distance measures are not applicable. For this reason, we employ the use of a DTW-based distance

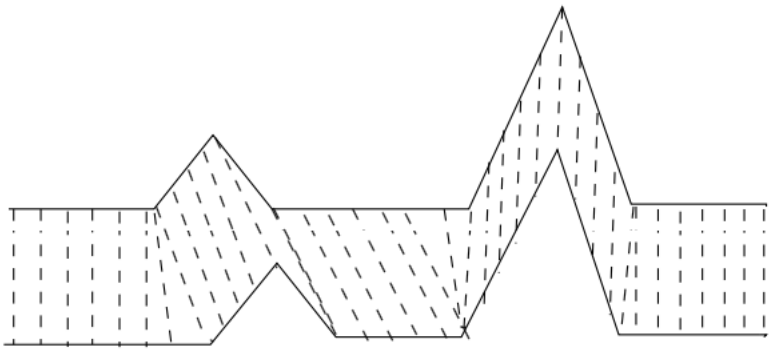


Figure 3.3: Similarity between two temporal signals via dynamic time warping. Obtained from *Wikipedia: Dynamic time warping.*

measure, which does not require temporal alignment nor synchronization between a pair of sequences in different sizes [79] (Figure 3.3).

Let $U = \{\mathbf{u}_{fj} | f = 1, \ldots, F_{\mathrm{U}}, j = 1, \ldots, J\}$ be the joint trajectories of an unknown action instance, with F and J as the total number of frames of the action and the number of joints, respectively. Note that length F_{U} of an unknown action and length F_i of an action instance in an AT are generally different. The local distance between the positions of joint j in frame f of U and frame f' in P_i is defined as the Euclidean distance as follows:

$$e(\mathbf{u}_{fj}, \mathbf{p}^i_{f'j}) = \|\mathbf{u}_{fj} - \mathbf{p}^i_{f'j}\|_2. \tag{3.2}$$

Then, using confidence value c_{fj} generated during the tracking we apply a mask to the trajectory of each joint j for each frame f. If this value is smaller than a predefined threshold τ, we determine that that part of the trajectory is not useful for classification. Therefore we assign a binary weight to each point of a joint trajectory by

$$w_{fj} = \begin{cases} 1 & \text{if } c_{fj} \geq \tau \\ 0 & \text{otherwise} \end{cases} . \tag{3.3}$$

This weighting is applied to the joint positions of both U and P_i. This means only J' out of the J joints are used for frame f, where J' is the number of joints

that are not masked ($J' \leq J$). Thus, we define the masked distance between all joint positions $\mathbf{u}_f$ and $\mathbf{p}^i_{f'}$ in frames f and f' as

$$d(\mathbf{u}_f, \mathbf{p}^i_{f'}) = \frac{1}{J'} \sum_{j=1}^{J} e(\mathbf{u}_{fj}, \mathbf{p}^i_{f'j}) w_{fj} w_{f'j}. \tag{3.4}$$

Using this distance, the DTW-based distance measure between U and P_i is defined as the minimum sum of the local distances over a warping path. Namely, letting $\mathbf{t}_n = (f_n, f'_n)$ be a pair of frames, f for the unknown action instance U and f' for the one in an AT, and $T = \{\mathbf{t}_n | n = 1, \ldots, N\}$ a warping path over which the sum is calculated, the DTW-based distance D is given by

$$D(U, P_i) = \min_T \sum_{(f_n, f'_n) \in T} d(\mathbf{u}_f, \mathbf{p}^i_{f'}) \tag{3.5}$$

$$\begin{aligned} \text{subject to} \quad & \mathbf{t}_1 = (1,1) \quad \text{and} \quad \mathbf{t}_N = (F_{\mathrm{U}}, F_i) \\ & f_1 = 1 \leq f_2 \leq \cdots \leq f_N = F_{\mathrm{U}} \\ & f'_1 = 1 \leq f'_2 \leq \cdots \leq f'_N = F_i \\ & \mathbf{t}_{n+1} - t_n \in \{(1,0), (0,1), (1,1)\}. \end{aligned} \tag{3.6}$$

Eq.(3.5) can be minimized by dynamic programming.

Since the nearest neighbor-based approach needs to compare the distances calculated for action instances of different length, a normalized version of this distance is calculated. The normalizing factor in this case is the length of the warping path T, that is

$$D'(U, P_i) = \frac{1}{N} D(U, P_i). \tag{3.7}$$

The action class m^* for the unknown action instance U is given as the one whose AT includes an action instance that gives the minimum distance with U, i.e.,

$$m^* = m_{i^*} \text{ where } i^* = \arg\min_i D'(U, P_i). \tag{3.8}$$

Algorithm 1 summarizes the action recognition process.

Algorithm 1 Proposed recognition method for Action Templates

Input: Unknown action sequence U
Initialize $\bar{D} = \infty$ **and** $\bar{i} = 0$
for each action instance P_i **in all ATs do**
  Calculate $D'(U, P_i)$
  if $D'(U, P_i) < \bar{D}$ **then**
    $\bar{D} \leftarrow D'(U, P_i)$
    $\bar{i} \leftarrow i$
  end if
end for
Return: $m_{\bar{i}}$

Figure 3.4: Example image of the datasets used. Left: self-generated, center: CMU MoCap, right: MSR-Action3D.

3.3 Experimental results

In order to evaluate our approach for generic HAR, we choose datasets containing heterogeneous actions [11] involving the whole body. More specifically, we used the CMU MoCap dataset, the MSR-Action3D dataset and our self-generated dataset, and compared the results with other state-of-the-art methods. A sample frame of each one is shown in Fig. 3.4.

3.3.1 Implementation details

The recognition algorithm was implemented in Matlab, running in Windows 8 (64 bit), installed in a PC with an Intel Core i7 processor and 16 GB RAM. In

addition, for the experiments, we used an empirically determined threshold value $\tau = 0.1$.

3.3.2 Self-generated UGOKI3D dataset

The UGOKI3D dataset was generated using a Microsoft Kinect v1 for evaluating our previous HAR method, which used the discrete Fourier transform and neural networks [59]. It is comprised of 8 heterogeneous actions that involve all body parts, and with different characteristics: periodic, aperiodic, static (the location of the user in the scene does not vary) and non-static. The actions are performed by 9 actors of different gender and appearances: (a) *bending*, (b) *jumping-jacks*, (c) *jumping-forward*, (d) *jumping*, (e) *side-galloping*, (f) *walking*, (g) *waving one hand*, (h) *waving both hands*. For the sake of comparability, we used the same evaluation scheme, applying leave-one-out (LOO) cross validation, in which we trained our model with sequences of 8 actors and evaluated our proposed method with the sequences of the remaining 1 actor. The accuracy was averaged over all 9 iterations.

The average accuracy rate obtained in this experiment was 94.44%, which is higher than the one achieved with our previous method (93.05%). The confusion matrix for all actions is shown in Table 3.1, whose rows and columns indicate the ground truth and recognition results respectively. As it can be observed, the most common classification errors involved actions that present similar fast position variations in the lower body, i.e. *jumping-forward* and *walking*. One of the reasons of these inaccuracies is the occasional errors in the skeleton tracking.

3.3.3 CMU MoCap dataset

To show the potential performance of our proposed method when the skeleton tracking is almost perfect, we used the motion capture dataset provided by Carnegie Mellon University, which contains actions captured at 120 fps [19]. This dataset was not generated from sequences captured with depth sensors, but with a motion capture technique using markers attached to the human body. This dataset is composed by multiple actors performing heterogeneous actions divided in categories such as locomotion and sports. However, not all the actors perform

Table 3.1: Confusion matrix for the UGOKI3D dataset

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
(a)	100% (9\9)							
(b)		88.89% (8\9)					11.11% (1\9)	
(c)			88.89% (8\9)		11.11% (1\9)			
(d)				100% (9\9)				
(e)					100% (9\9)			
(f)			11.11% (1\9)		11.11% (1\9)	77.78% (7\9)		
(g)							100% (9\9)	
(h)								100% (9\9)

every action, and the number of instances of each action can vary largely. To be consistent with the experiment in the previous section, a subset of 8 different actions was selected, with a noticeable emphasis on the lower body, i.e. (a) *running*, (b) *walking*, (c) *jumping forward*, (d) *jumping*, (e) *soccer kick*, (f) *boxing*, (g) *jumping jacks*, (h) *hand signs*. Also, although the dataset offers joint trajectories in more than 20 body parts we use its subset that corresponds to the 15 joints of our UGOKI3D dataset. In addition, since this skeleton tracking method does not provide a confidence parameter, we did not use masking for this experiment ($w_{fj} = 1$).

Our method was evaluated applying LOO cross validation again, achieving the accuracy of 97.22%. The accuracy for each action is summarized in the confusion matrix of Table 3.2. Only the *jumping jacks* action is misclassified twice; in one sequence the actor only performed half a repetition, and in the other the actor did

not move the arms accordingly to the action. As expected, due to the accurate joint estimates, the results of this experiments were highly accurate, regardless of the types of actions. We also evaluated our previous method [59], resulting in an inferior accuracy of 91.67%.

Table 3.2: Confusion matrix for the CMU MoCap dataset

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
(a)	100% (9\9)							
(b)		100% (9\9)						
(c)			100% (9\9)					
(d)				100% (9\9)				
(e)					100% (9\9)			
(f)						100% (9\9)		
(g)				11.11% (1\9)			77.78% (7\9)	11.11% (1\9)
(h)								100% (9\9)

3.3.4 MSR-Action3D dataset

The MSR-Action3D dataset includes various challenging actions and has been widely used to evaluate HAR methods. This dataset contains twenty different static actions performed by up to 10 actors, and the same actor did the same action from one to three times. The actions are: (a) *high arm wave*, (b) *horizontal arm wave*, (c) *hammer*, (d) *hand catch*, (e) *forward punch*, (f) *high throw*, (g) *draw x*, (h) *draw tick*, (i) *draw circle*, (j) *hand clap*, (k) *two hand wave*, (l) *side-boxing*, (m) *bend*, (n) *forward kick*, (o) *side kick*, (p) *jogging*, (q) *tennis swing*, (r) *tennis*

serve, (s) *golf swing*, (t) *pickup & throw*. The dataset was built using sequences captured with depth sensors at 15 fps. It provides the 3D position and the tracking confidence of 20 joints per frame, but we kept using 15 joints for our proposed method since we considered the extra five (wrists, ankles, and center hip) do not add much information to the model. Although some works highlight its difficulty resides in the similarity of its actions, in our opinion, the dataset is challenging due to the noise present in the skeleton tracking. Fig. 3.5 shows some unrealistic poses included in the dataset.

Table 3.3: Action subdivision of the MSR-Action3D dataset used in the experiments

Subset 1 (SS1)	Subset 2 (SS2)	Subset 3 (SS3)
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup & throw

We followed the evaluation methodology employed in previous works [4, 48, 73, 107], and divided the 555 instances into three groups as shown in Table 3.3. For each group, we conducted a cross-subject experiment in which the actions performed by actors 1, 3, 5, 7, and 9 were used for training and the ones from actors 2, 4, 6, 8, and 10 for testing. The overall recognition accuracy obtained in the experiment was 84.09%. The individual accuracy rates for SS1, SS2, and SS3 are 80%, 78.57%, and 93.69% , respectively. The first two subgroups were more erroneous than the third one. These results are shown in detail in Tables 3.4, 3.5, and 3.6.

Table 3.7, obtained partially from [107], shows the generalization performance of our method compared with other state-of-the-art methods that were evaluated

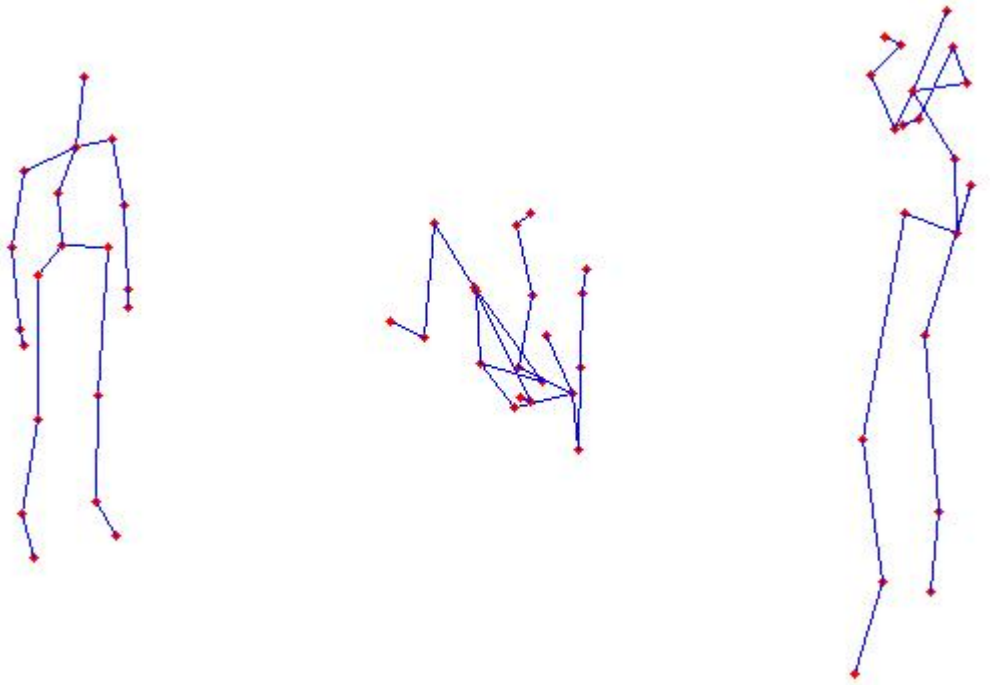


Figure 3.5: Noise present in the MSR-3DAction dataset. Left: base position, center and right: noisy samples.

against this dataset using the same configuration. The upper part of the table lists the methods that are capable of runtime learning (e.g. NN), and the lower part of the table lists the ones that are not (e.g. SVM). Our method's accuracy outperforms the other HAR methods that are capable of runtime learning by far, and is very close to the state-of-the-art methods. Compared with the other two datasets used, the MSR-Action3D has a larger presence of tracking noise. As Müller and Röder remarked in [66], when performing HAR with noisy templates, recognizing new actions becomes hard (see Table 3.5). However, when we apply the confidence value of the skeleton tracker to avoid using the erroneous sections in the AT, matching the recognition performance of our method improves noticeably, as shown in Table 3.7.

Table 3.4: Confusion matrix for the MSR-Action3D dataset (SS1)

	(b)	(c)	(e)	(f)	(j)	(m)	(r)	(t)
(b)	50% (6\12)	8.33% (1\12)	41.67% (5\12)					
(c)		75% (9\12)	25% (3\12)					
(e)			100% (11\11)					
(f)	18.18% (2\11)	9.09% (1\11)		72.73% (8\11)				
(j)					100% (15\15)			
(m)						46.67% (7\15)		53.33% (8\15)
(r)							100% (15\15)	
(t)							7.14% (1\14)	92.86% (13\14)

3.3.5 Flexible HAR

We evaluate the performance of our proposed method's capability of learning new action instances in runtime. We assume a scenario of a customizable gesture interface for a certain application system, in which a command for the system is issued via the gesture interface whose backend is our HAR method. This scenario supposes that the gesture interface has a predefined set of gestures, each of which has a single instance of the corresponding gesture when initialized. The interface learns at runtime; if the interface fails in correctly recognizing an input instance of a gesture, the user specifies the correct label of the instance and the interface includes it to the corresponding AT.

To demonstrate the performance under this scenario, we used the action classes contained in each subset of the MSR-Action3D dataset instead of actual gestures (8 different action classes per subset). We used 20 action instances of each action

Table 3.5: Confusion matrix for the MSR-Action3D dataset (SS2)

	(a)	(d)	(g)	(h)	(i)	(k)	(l)	(n)
(a)	83.33% (10\12)	8.33% (1\12)	8.33% (1\12)					
(d)	50% (6\12)	16.67% (2\12)	16.67% (2\12)				16.67% (2\12)	
(g)			92.31% (12\13)	7.69% (1\13)				
(h)	20% (3\15)			80% (12\15)				
(i)	26.67% (4\15)		13.33% (2\15)		60% (9\15)			
(k)						100% (15\15)		
(l)		6.66% (1\15)					86.68% (13\15)	6.66% (1\15)
(n)								100% (15\15)

class in the subset, and divided it into two groups: 10 for learning and 10 for testing. That is, for each subset we use a learning and testing groups of 80 action instances each. At the start, we generate the ATs with a single instance for each class, and then we feed the remaining instances in the learning group one by one (72 instances in total). If our HAR method fails to recognize one instance, it adds that instance to the corresponding AT. We evaluated the accuracy of the method using the test set after an instance in the learning group is input. We repeat this 100 times, randomizing the instances in the learning and testing groups, and the order of the input learning instances. The recognition accuracy is the average of all repetitions. We also measured the time required for recognizing the instances in the test set, which is also averaged over the 100 repetitions.

Figure 3.6 shows the runtime accuracy of our method for each instance in the learning group evaluated against the test group. The final recognition accuracies

Table 3.6: Confusion matrix for MSR-Action3D dataset (SS3)

	(f)	(n)	(o)	(p)	(q)	(r)	(s)	(t)
(f)	81.82% (9\11)			18.18% (2\11)				
(n)		100% (15\15)						
(o)			90.91% (10\11)	9.09% (1\11)				
(p)				100% (15\15)				
(q)					100% (15\15)			
(r)						100% (15\15)		
(s)							100% (15\15)	
(t)					28.57% (4\14)			71.43% (10\14)

achieved for subsets SS1, SS2, and SS3 are 75.12%, 79.06%, and 88% respectively, with 37, 35, and 27 instances on average added to the ATs respectively (see Figure 3.7). By comparing these results to the previous experiment, it can be noticed that our method is able to provide a similar accuracy generating ATs in runtime with less than half the action instances than the previous configuration. It is also remarkable the fact that our method achieves accuracies around 50% with just a single instance per action class. Figure 3.8 shows the time in seconds spent in classifying one gesture using our implementation. It grows from 0.5 sec to about 2 sec almost linearly as the number of learned instances in our ATs grows.

Table 3.7: Recognition accuracy comparison for the MSR-Action3D dataset

Method	Accuracy	Type
Proposed method	**84.09%**	**Skeleton**
Proposed method (no noise masking)	**79.31%**	**Skeleton**
Rate-invariant Analysis (NN) [4]	63%	Skeleton
Dynamic Temporal Warping [66]	54%	Skeleton
MMTW [108]	92.57%	Skeleton
Joint Movement Similarities [75]	91.2%	Skeleton
HOPC [80]	90.9%	Depth
Rate-invariant Analysis (SVM) [4]	89%	Skeleton
HON4D [73]	88.36%	Depth
Mining Actionlet Ensemble [107]	88.2%	Skeleton
Histograms of 3D joints [113]	78.97%	Skeleton
Action Graph on Bag of 3D Points [48]	74.7%	Depth
Hidden Markov Model [54]	63%	Skeleton
Recurrent Neural Network [58]	42.5%	Skeleton

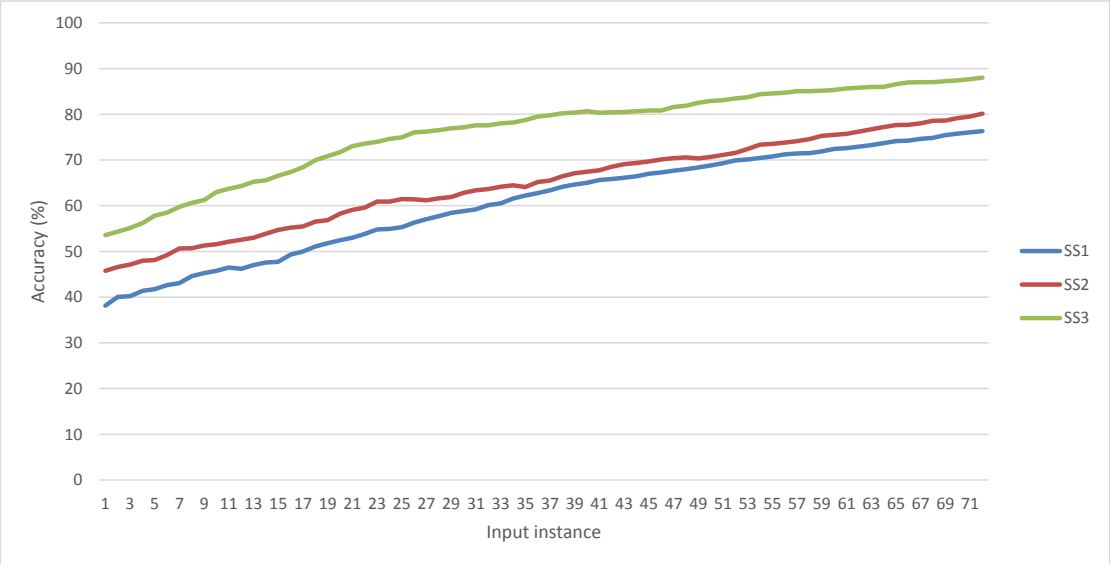


Figure 3.6: Recognition accuracy during runtime learning. Horizontal axis: input instances, vertical axis: average recognition accuracy (percentage).

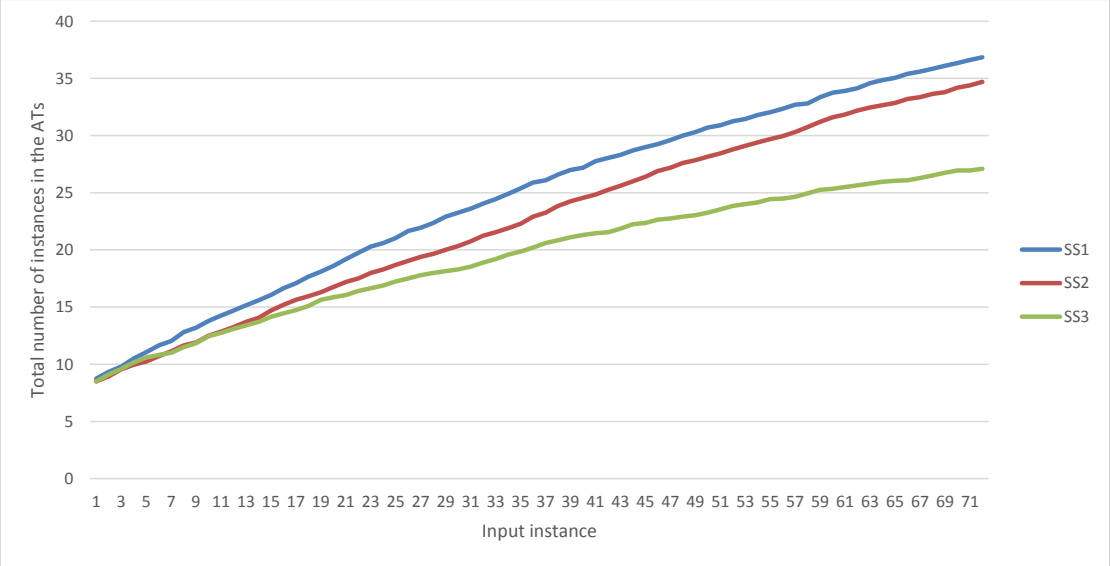


Figure 3.7: Total number of instances in the ATs during runtime learning. Horizontal axis: input instances, vertical axis: average number of instances in the ATs.

3.3.6 Discussion

Our experimental results have shown that our approach can be successfully applied for HAR at runtime in depth video sequences. In comparison with many related works, we use raw 3D joint trajectories instead of other representations [59, 66, 107] such as Fourier transform, joint mining, or boolean features, thereby reducing the computational cost of learning. By applying DTW we gain robustness against variations in execution rates, which heavily affect HAR. Although this methodology is more sensitive to the noise present in the joint position estimation, we manage to effectively alleviate this problem by using the confidence values provided by the skeleton tracker itself. We achieved high recognition rates for a wide variety of actions (periodic, static, etc.) and sensors (high frame rate, low frame rate), and outperformed other methods that are capable of runtime learning on the challenging MSR-Action3D dataset.

Compared to the state-of-the-art methods that are not capable of runtime learning, our performance is slightly inferior. We consider the reason is that we do not rely on an intricate training phase in order to reduce the cost of learning a

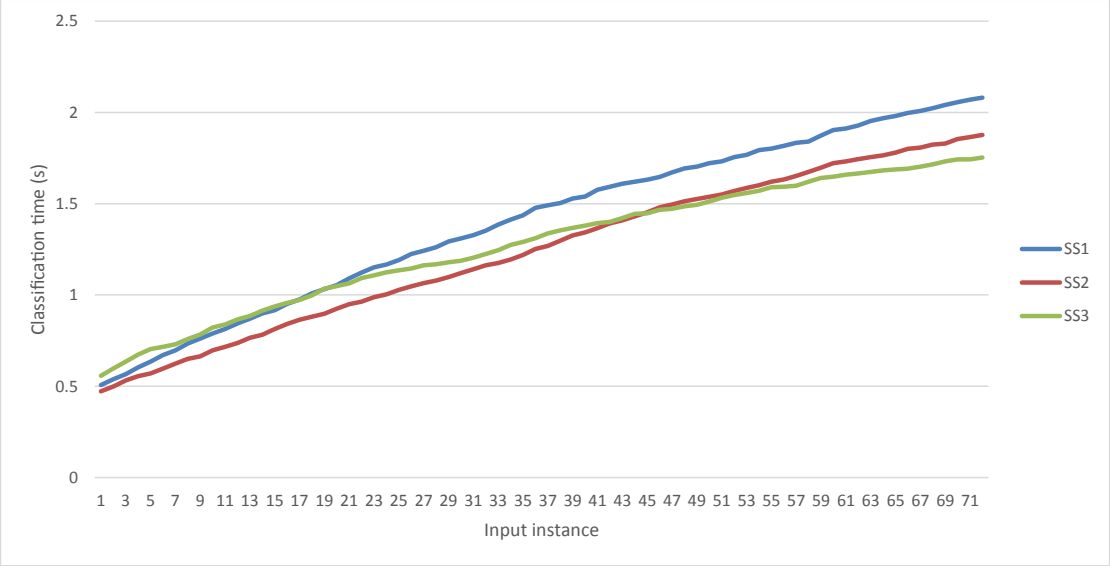


Figure 3.8: Classification time for one instance during runtime learning. Horizontal axis: input instances, vertical axis: average classification time (seconds).

new action instance. Besides, our feature set consists of a small number of joint positions tracked in real time, with no other RGB/depth information. Basically, our method deals with a trade-off between flexibility and accuracy in order to allow for runtime learning. For example, Wang and Wu [108] also deal with variations in execution rate of actions using a human joint model. But contrary to our proposed method, their maximum margin temporal warping (MMTW) method relies on a costly SVM algorithm in order to extract the optimal template for the training dataset. Therefore, it can be considered that MMTW is not suitable for runtime learning of new action instances. Also, in their skeleton approach, they use 1140 ($20 \times 19 \times 3$) features per frame which is the distance in the three-dimensional space of each body joint offered by Kinect to the rest. In order to maintain the computational efficiency, our method uses only 45 (15×3) features per frame. The same can be said for other works [4, 75, 107].

Besides, we have proved experimentally that our method offers a great flexibility that would allow users to provide some feedback on wrong classifications or even to add a new action category at runtime. Also, an AT can contain instances

for several ways of performing the same action class (e.g. drinking with your left hand or right hand, gesturing standing up or sitting, etc.), which provides robustness against variations in the way actions are executed. To the best of our knowledge, this feature is not present in the other methods that, in spite of achieving a higher recognition rate in noisy conditions, suffer from a computationally expensive and intricate learning phase demanding a large amount of training data. In our method, the computational complexity of learning one action is $O(f \times j)$, where f is the number of frames the action lasts for and j is the number of joints tracked. Table 3.8 contains the learning and classification times of our method for each dataset, using a Matlab implementation and the computer equipment described in Section 3.3.1. Another example of its flexibility is that, in case of performing action recognition of a specific body part, the number of trajectories used can be easily modified, generating customized ATs with just the joints of interest (hands, legs, etc.). Also, the joint positions contained in an AT itself can be used to reproduce the captured action, which is useful for animation purposes.

Table 3.8: Learning and classification times for each dataset

Dataset	Learning time (training set)	Average classification time (1 action)
UGOKI3D: 8 actions (64 frames, 9 actors)	0.24s	8.9s
CMU MoCap: 8 actions (150∼800 frames, 9 actors)	0.56s	219.5s
MSR Action3D: 8 actions/subset (50∼80 frames, 12∼15 actors)	0.64s	7.13s

This work also shows an effective way for applying DTW to action recognition. To the best of our knowledge, the previous results of using exclusively DTW in a 3D joints-based HAR methods have not been convincing enough [85]. Although intuitively DTW fits quite well a task such as analyzing action trajectories, it has been criticized arguing that it is more sensitive to temporal scale changes than HMM-based methods [54], and produces large temporal misalignments in case of

periodic actions [107]. But rather than that, by looking at the actions used in the experiments and the results obtained, we inferred that what most affects this technique was the noise in the skeleton tracking process.

When the number of instances in our ATs increases, the time cost of our method in order to classify one action can be high (Figure 3.8) due to the computational cost of DTW, $O(MN)$, where M and N are the lengths of the two compared sequences [3]. However, implementing a real-time system would not be infeasible due to the increasing speed of computers and acceleration techniques based on parallel execution such as GPGPU, given the fact that in our algorithm distance calculations can be executed completely in parallel. Our method has also the advantage of not requiring a large number of action instances.

3.4 Summary

In this chapter, we have presented a flexible method for recognizing actions from trajectories estimated from depth sequences based on the generation of action templates using joint trajectories. To deal with inaccuracies in the joint position estimation, our method integrates a mask for the noisy sections of the trajectories during classification using the confidence values offered by the 3D joint position estimation algorithm [120]. The proposed method deals with a trade-off between flexibility and accuracy, achieving comparable results with the state-of-the-art methods in a challenging dataset. We have also successfully demonstrated the flexibility of our approach, which allows performing HAR with very few training instances, while learning new actions at runtime. This is a very powerful feature in applications such as action databases, video analysis, and customizable gesture interfaces.

The contributions of this work are summarized as follows:

- We proposed a novel method for flexible HAR that allows updating the action classifiers at runtime and classification with few training instances.
- We also proposed a modification of the classification algorithm to mask noisy joint trajectories by using the confidence values from the skeleton tracker.

- We evaluated experimentally the performance of our method and its adequacy for runtime learning of actions in depth sequences. The results demonstrate the effectiveness and accuracy of our method along with its flexibility.

4 Summarization of user-generated sports video based on HAR results

4.1 Overview

In Chap. 3 we introduced an action recognition method based in template matching of actions that can be applied to recognition problems that do not have a large number of training instances. It works with body joints in 3D estimated from the depth maps in RGB-D video. Our intention is to use the recognition results of this method to model the interesting parts of a user-generated sports video. As explained in 2.1.1, for UGV we cannot use the same methods as other works in sports video summarization, so our novel idea is that the players, a constant element in a sports video of any kind, can be used as a source of features for summarization by recognizing their actions. In order to test our hypothesis, we recorded our own dataset of an example sport (Kendo, or Japanese fencing) using and RGB-D camera. The reason we used depth information is to ensure the actions of the players were properly recognized, so highlights can be modeled better. This chapter explains the methodology of this approach in detail and the first results ever in HAR-based summarization of user-generated sports video.

4.2 Recognizing players' actions for summarization UGV of sports

Figure 4.1 depicts an overview of our method, which takes an RGB-D sports video sequence and generates a summary containing the highlights of the game. The sequence is firstly segmented into T uniform-length (*i.e.*, 3 seconds) sub-sequences. In order to exploit the inherent semantics of the video, we apply HAR to each sub-sequence. In most sports, multiple players are involved in the game; therefore, HAR is also applied to each player to calculate the dissimilarity between the action of that player in each sub-sequence and each action instance in a predefined set of action classes. We use this dissimilarity and an activity measure, which quantifies the amount of motion in the sub-sequence, to model interesting sub-sequences that are to be included in the resulting highlights summary with a hidden Markov model with Gaussian mixture model emissions (GMM-HMM), which is trained with labeled sub-sequences. Finally the summary is extracted via skimming curve formulation [102] for a given time length L.

4.2.1 HAR via Action Templates

In order to calculate the dissimilarity between the action of players in the t-th sub-sequence and each of the predefined actions, we apply HAR to each player p. From the depth maps in a sub-sequence, we obtain the skeleton (*i.e.*, a set of 3D joint positions) of each player using a skeleton tracker ([120], for example) to gain robustness to view variations with respect to both the camera locations and subject appearances. We use a simple method for HAR [24], which calculates the distance between the sequence of skeletons of player p in a sub-sequence and each of the action templates (referred to as ATs) in an action dataset.

An AT is a set of action instances (sequences of skeletons) of a predefined action class specialized for the sport. To generate an AT, we extract the skeleton from a depth map sequence that contains one of the predefined actions. Skeleton trackers can also provide a confidence value for each estimated joint position. These positions are transformed to the player's coordinate system, whose origin is at one of the joints (*e.g.*, torso). The sequence of transformed skeletons along with the confidence values form the AT.

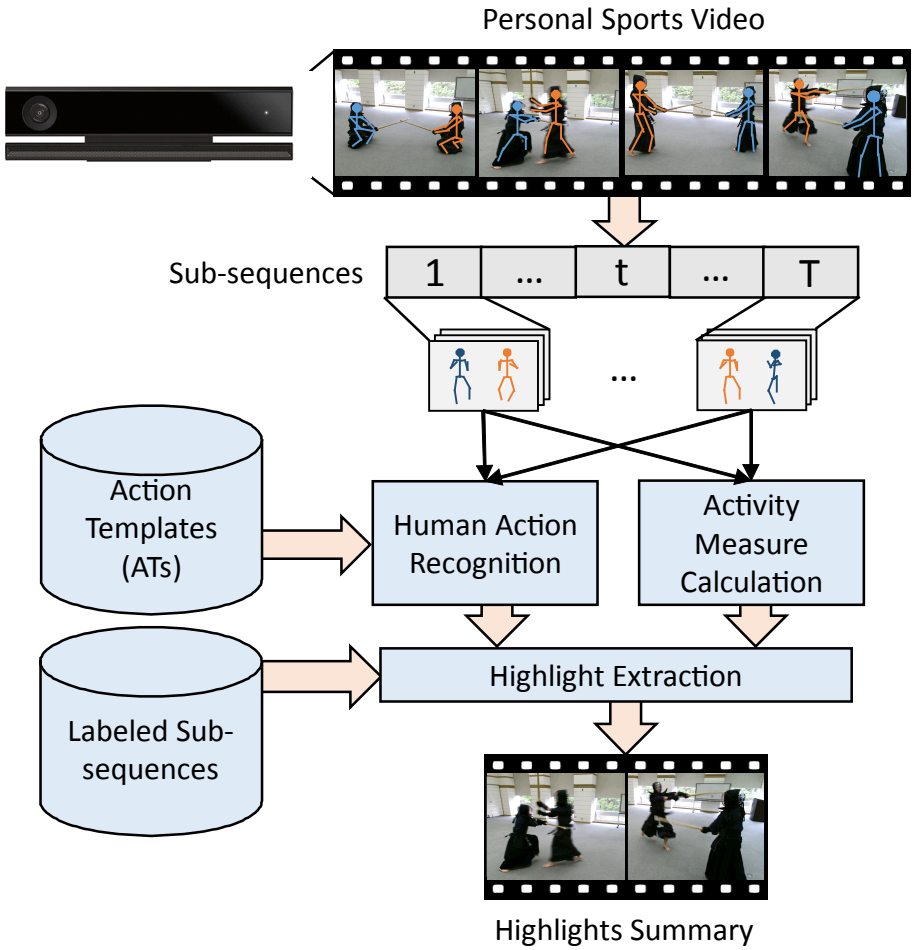


Figure 4.1: Overview of our summarization method.

For the given t-th input sub-sequence, which may contain multiple players in unknown action classes, we apply a similar process to extract the players' skeletons and transform them into each player's coordinate system. We then calculate the distance between the sequence of skeletons for each player and each of the ATs. Since the duration of an action varies from instance to instance, we adopt dynamic time warping [79] to handle this. In this method, the confidence values are used to filter the noisy sections of the trajectories. Let N denote the number of the predefined actions classes and M the number of action instances per action class. Our HAR method generates a vector $\mathbf{d}_{tp}$ whose n-th element d_{tp}^{n} is given by $d_{tp}^{n} = \min_m d_{tp}^{nm}$, where d_{tp}^{nm} is the distance between player p's action

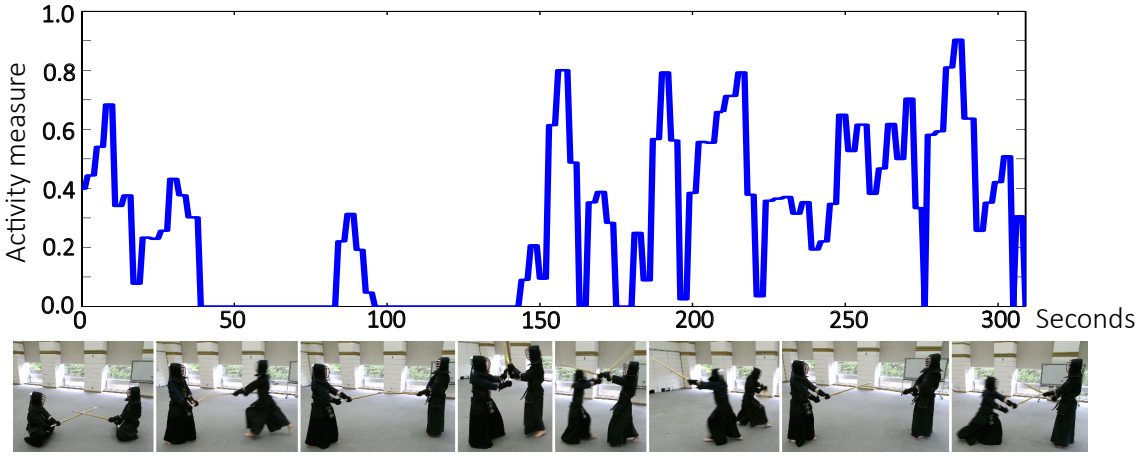


Figure 4.2: Activity measure along the course of a Kendo game.

in t-th sub-sequence and the m-th AT for the n-th action class ($m = 1, \ldots, M$ and $n = 1, \ldots, N$).

4.2.2 Activity measure

The HAR outputs may not reflect how sudden or prominent the actions are. In [12], they hypothesize that interesting highlights in sports video are characterized by certain patterns in the entropy of the intensities in RGB frames. For each sub-sequence, we use the activity measure of each player's motion based on the entropy of the motion of each joint. For this, we divide the 3D space of the player's coordinate system into V volumes and calculate the ratio r_v of the number of frames in the subsequence in which the joint j of player p fall into volume v. The entropy for joint j is given by

$$e_j = -\sum_{v=1}^{V} r_v \log(r_v). \tag{4.1}$$

We define the activity measure of a player as $a = \sum_{j=1}^{J} e_j$ where J is the total number of joints. Figure 4.2 shows the variation of a along time. The activity measure rises as sudden actions are executed successively, and decreases with repetitive motion (or lack of motion). Sections with zero activity are those where players were not recognized.

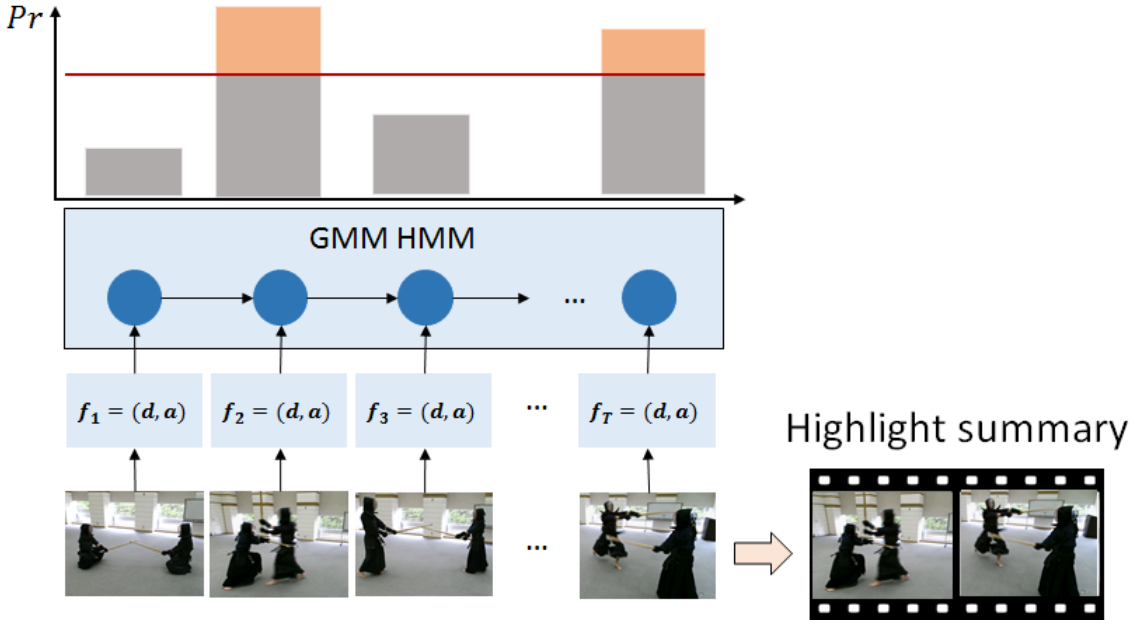


Figure 4.3: The probability of each segment being interesting (Pr) is calculated using a GMM-HMM that models the temporal relationships between the calculated features. We obtain the highlight summary by thresholding Pr.

For sub-sequence t, we define a feature vector $\mathbf{f}_t^\top = (\mathbf{d}_{t1}^\top, a_{t1}, \mathbf{d}_{t2}^\top, a_{t2}, \ldots, \mathbf{d}_{tP}^\top, a_{tP})$, which is a concatenation of the HAR result $\mathbf{d}_{tp}$ and activity measure a_{tp} for all players, where P is the number of the players in the t-th sub-sequence and a_{tp} is the activity measure for player p.

4.2.3 Highlight extraction

In order to create the summary from the original sequence, we calculate the probability of each sub-sequence of being interesting/non-interesting based on the features, assuming that the segments that are labeled as interesting by users are the highlights of the game. We adopt a GMM-HMM [6] to model interesting/non-interesting segments because adjacent sub-sequences are expected to be highly correlated. Figure 4.3 shows an overview of the highlight extraction process.

In our method, we assume that the emission probability $\Pr(\mathbf{f}_t|e)$ of $\mathbf{f}_t$ given e

follows a Gaussian mixture model, where $e = 1$ indicates that the sub-sequence belongs to an interesting segment and $e = 0$ otherwise. Specifically, the emission probability is given by

$$\Pr(\mathbf{f}_t|e) = \sum_{k=1}^{K} w_{ek}\mathcal{N}(\mathbf{f}_t|\mu_{ek}, \mathbf{\Sigma}_{ek}), \tag{4.2}$$

where w_{ek}, μ_{ek}, and Σ_{ek} are the mixture weight, the mean, and the covariance matrix of the k-th mixture component for state e. Letting $F = \{\mathbf{f}_t|t = 1, \ldots, T\}$ and $\mathbf{e}^\top = (e_1, \ldots, e_T)$, the probability $\Pr(F_T, \mathbf{e})$ is given by

$$\Pr(F, \mathbf{e}) = \Pr(e_0) \prod_{t=1}^{T} \Pr(e_t|e_{t-1}) \prod_{t=1}^{T} \Pr(\mathbf{f}_t|\mathbf{e}_t, \phi), \tag{4.3}$$

where $\Pr(e_0)$ is the initial state probability. We can calculate the posterior probability $\Pr(e_t|F)$ using the forward-backward algorithm. Since we have labeled videos for training, the parameters for initial state probability $\Pr(e_1)$ and the transition probability $\Pr(e_t|e_{t-1})$ can be easily determined by counting, and the parameters for GMM (*i.e.*, w_{ek}, μ_{ek}, and Σ_{ek}) can be estimated using the EM algorithm [64].

Once the probabilities are obtained, we generate the summary using skimming curve formulation [102]. Given a certain summary length L in seconds, we apply thresholding to $\Pr(e_t|F)$ by reducing the threshold until we find a set of segments whose total length in seconds is the largest below L. We arrange the extracted segments in temporal order to generate a video summary. Algorithm 2 shows this process.

4.3 Experimental results

4.3.1 Implementation details

To evaluate our method, we chose Kendo as an example sport, which is a martial art featuring two players and a set of recognizable actions. Using a Microsoft Kinect v2 sensor, we recorded 10 RGB-D videos (90 minutes in total), which contain 12 combats. The videos used in the experiments were taken close to the players (2m–4m) for depth map acquisition. We used [120] for skeleton tracking.

Algorithm 2 Highlight extraction by thresholding $\Pr(e_t|F)$

```
Empty highlights H = ∅
For a given summary length L and subsequence length l
Initialize threshold = 1 and δ = 10^−5
while L − l ≥ 0 do
  for each subsequence t in T do
    if L − l < 0 then
      stop iterating T
    else if Pr(e_t|F) >= threshold then
      H ∪ {t}
      L = L − l
    end if
  end for
  threshold = threshold − δ
end while
Sort H by t
```

Apart from these videos, we generated a dataset for HAR, which contains 200 action instances (10 action classes×4 actors×5 repetitions) of action classes (a) *men*, (b) *kote*, (c) *dou*, (d) *bougyo*, (e) *kamae*, (f) *tsubazeriai*, (g) *hikimen*, (h) *sonkyo*, (i) *osametou*, and (j) *aruki*. These actions consist of strikes in different body parts and defense positions * (Fig. 4.4). We evaluated the used HAR method with this dataset in the leave-one-out (LOO) fashion. Table 4.1 shows the recognition results for each action class. The high-speed of the actions and players' clothes hindered HAR, and similar actions were often mistaken. Its generalization performance is evaluated in [24] against the MSRAction3D dataset with the configuration used in [48]. The used method has an accuracy of 84.1%, surpassing [48] (74.7%), and other nearest neighbors-based methods [4] (63%). However, this accuracy is a bit lower than that of methods with a more costly training, such as support vector machines [107] (88.2%), [75] (91.2%) or convolutional neural networks [110] (94.6%).

We asked 13 participants to evaluate our method. Since the interestingness of

*A description can be found at https://en.wikipedia.org/wiki/Kendo

Table 4.1: Confusion matrix of [24] over the kendo dataset (%).

		Recognition results									
		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
Action classes	(a)	25	20	15	5	25			10		
	(b)		20	30	20	5	10				15
	(c)	15	10	50	5	10			5		5
	(d)	10		5	15	45	25				
	(e)	20			20	40	20				
	(f)	10			35	35	20				
	(g)	20		5				50		25	
	(h)	60	10						30		
	(i)	35				5		10	5	45	
	(j)	50	20	5						10	15

the extracted highlights can differ from one user to another, we grouped them into experienced (E) and non-experienced (NE) in Kendo, which would affect the results the most. Group E has 3 users and NE has 10. In order to train the GMM-HMM for highlight extraction, 3 and 5 users from groups E and NE were employed as annotators, and assigned interesting/non-interesting labels to the sub-sequences in the 10 original videos. Each sub-sequence was judged to be interesting if two or more annotators labeled it as interesting. Whereas group E picked sub-sequences with very specific actions (*e.g.*, very fast strikes, decisive strikes, etc.), group NE picked a more general set of actions (*e.g.*, non-decisive strikes, feints, etc.), reaching about twice the number of sub-sequences than group E. Again in the LOO fashion, we trained the GMM-HMM with the labels of 9 videos to generate the summary of the remaining.

4.3.2 GMM-HMM objective evaluation

We evaluated the performance of our trained GMM-HMM by thresholding $\Pr(e_t = 1|F) > 0.5$, and calculating precision (P), recall (R), and f-score (F) metrics for the extracted sub-sequences. Due to the limitations of the capturing device, in some parts of the original video, one or both players were not recognized. For this

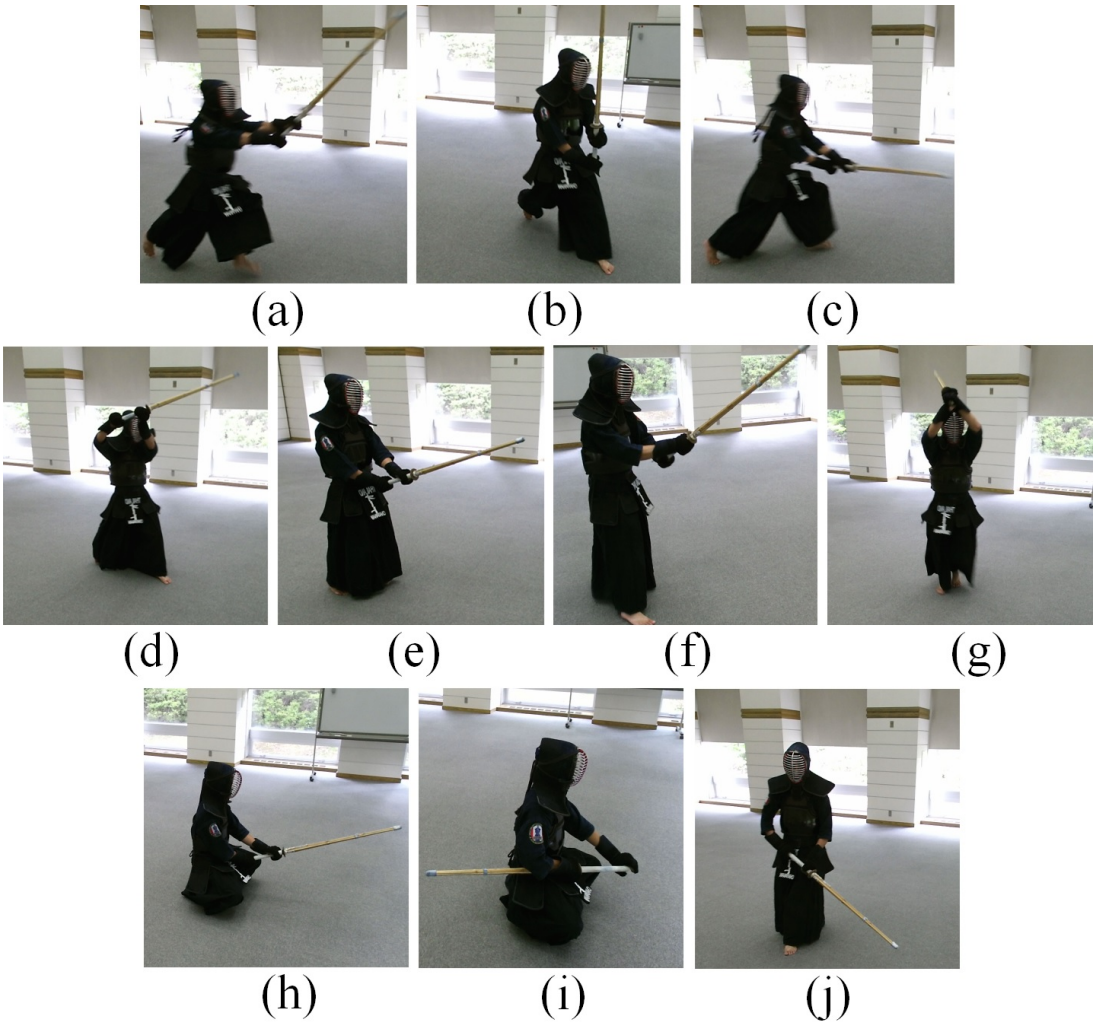


Figure 4.4: Actions used in the dataset.

reason, we evaluated the performance under these conditions: all sub-sequences (A, B) and only the sub-sequences in which both players' skeleton is tracked (C, D). We also evaluated the difference in performance when the activity measure is used (A, C) or not (B, D). Table 4.2 shows the results. The best results correspond to the case where both players' skeletons were tracked and activity measure was used (C). The effect of including our activity measure is greater on group *E*'s results. Since group *E*'s annotations included more specific actions, it seems the activity measure helps to discern specific interesting actions among similar HAR

results. When comparing groups *E* and *NE*, the latter's performance is higher since their annotations contain a broader set of actions.

Table 4.2: GMM-HMM performance.

	Annot. *E*			Annot. *NE*		
	P	R	F	P	R	F
(A)	0.41	0.44	0.42	0.62	0.76	0.68
(B)	0.39	0.42	0.41	0.62	0.75	0.68
(C)	**0.57**	**0.72**	**0.63**	**0.79**	**0.77**	**0.78**
(D)	0.49	0.64	0.56	0.77	0.75	0.76

4.3.3 Video summary objective evaluation

Our generated summaries are composed of sub-sequences with their estimated labels of interestingness. Human annotators expected that a set of consecutive sub-sequences with interest labels (referred to as a highlights, hereinafter) contain an event in a certain granularity. Therefore, even a single missed sub-sequence in the set may distract viewers. For this, we objectively evaluated our method by modifying the definitions of precision and recall to take into account the completeness of the extracted highlights. We define the completeness criterion for an extracted highlight as the fraction of overlap with its associated highlight from the ground truth annotated by our participants. Associating extracted and ground truth highlights is not trivial, and we did this in a greedy manner, in which the total number of overlapping sub-sequences is maximized. We deemed an extracted highlight as a true positive (TP) if it covers over $C\%$ of the sub-sequences in the associated ground truth highlight. In this experiment, we thresholded $\Pr(e_t|F_T)$ in the range $[0, 1]$ (instead of 0.5 as in section 4.1) to generate summaries of different lengths.

Figure 4.5 shows the recall-precision curves produced for $C = 50\%$, 70%, 90%. Whereas almost all highlights with $C = 70\%$ reached also $C = 90\%$, when reducing C to 50% the number of TP increases significantly. We attribute the presence of incomplete segments to the transition probabilities of our GMM-HMM model,

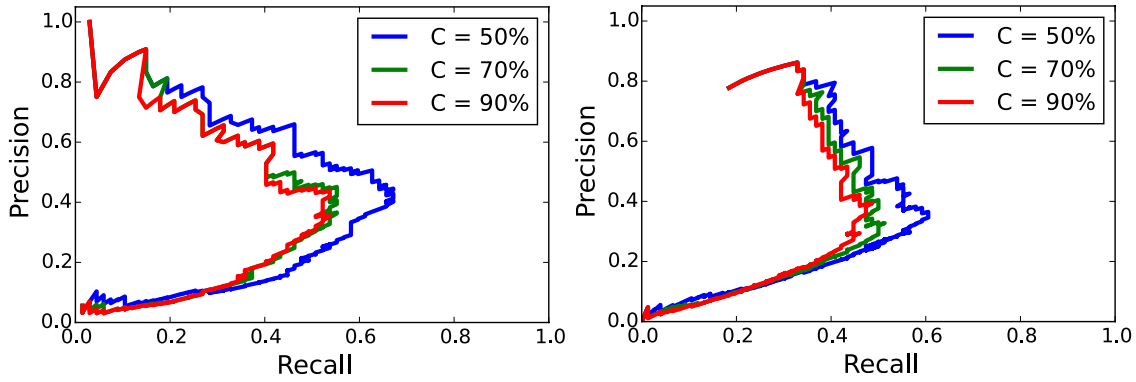


Figure 4.5: Recall-precision curves for groups E (left) and NE (right)

which are very low for the *non-interesting to interesting* transition and higher for the *interesting to non-interesting* one. This makes highlights start later and begin earlier than the annotated ground truth. When comparing groups E and NE, the latter's recall shows a higher and more constant number of TPs for different summary lengths, which is consistent with the results shown in section 4.1. We conclude that our method is able to detect very well certain highlights, but others remain incomplete.

4.3.4 Video summary subjective evaluation

We assessed the quality and usefulness of our video summaries from the users' point of view by means of a survey. All 13 participants watched the video summaries that, for $C = 70\%$, gave the (a) maximum, (b) median, and (c) minimum f-scores averaged for groups E and NE in the previous section, as well as their corresponding original video. We also used different summary lengths $L = 20$, 30, and 40 s, to see how the length affects viewers' perception. For comparison, besides the summaries created with groups E and NE annotations, we also evaluated video summaries based on the k-means clustering algorithm as a baseline, in which clustering was performed on our HAR features. As a result, every participant watched 27 summaries.

We asked participants (Q1) if each summary showed an entire action from beginning to end, (Q2) if each summary was interesting, (Q3) if the participant got an insight on the original video by watching the summary, and (Q4) if the

Table 4.3: Survey results according to the summary type. Each cell consists of the mean ± standard deviation of the subjective scores.

		Summary type		
		Annotations *E*	Annotations *NE*	Clustering
Q1	Group *E*	**3.44±0.67**	3.04±0.72	1.89±0.69
	Group *NE*	**3.63±0.5**	**3.63±0.49**	2.26±0.78
Q2	Group *E*	**3.33±0.58**	3±0.33	1.37±0.35
	Group *NE*	**3.79±0.53**	3.78±0.3	1.88±0.55
Q3	Group *E*	**3.33±0.33**	3.11±0.58	1.33±0.29
	Group *NE*	3.57±0.54	**3.68±0.39**	1.92±0.49
Q4	Group *E*	4.41±0.57	**4.67±0.33**	2.22±0.58
	Group *NE*	3.6±0.34	**3.62±0.36**	2.27±0.35

Table 4.4: Survey results according to the summary lenght. Each cell consists of the mean ± standard deviation of the subjective scores.

		Summary length		
		20 s	30 s	40 s
Q1	Group *E*	3±0.7	**3.56±0.58**	3.17±0.81
	Group *NE*	3.58±0.46	**3.75±0.43**	3.57±0.61
Q2	Group *E*	2.89±0.62	**3.33±0.21**	3.28±0.49
	Group *NE*	3.53±0.5	**3.92±0.32**	3.9±0.36
Q3	Group *E*	3.11±0.66	**3.33±0.21**	3.22±0.5
	Group *NE*	3.38±0.48	**3.77±0.38**	3.72±0.49
Q4	Group *E*	4.44±0.69	**4.61±0.44**	4.56±0.27
	Group *NE*	3.47±0.41	**3.8±0.27**	3.57±0.29

summary was not redundant. Tables 4.3, 4.4 and 4.5 show the results for each question. Answers are averaged for group *E* and *NE* separately and grouped by the summary type, length, and video. The latter two cover the answers for summaries created with annotations *E* and *NE* together. By looking at the first row,

Table 4.5: Survey results according to the f1score of the video. Each cell consists of the mean ± standard deviation of the subjective scores.

		Video f1score		
		Highest	Median	Lowest
Q1	Group *E*	**3.61±0.88**	3.11±0.58	3±0.56
	Group *NE*	**3.9±0.54**	3.75±0.35	3.25±0.33
Q2	Group *E*	**3.28±0.57**	**3.28±0.39**	2.94±0.49
	Group *NE*	**4.1±0.29**	3.8±0.24	3.45±0.45
Q3	Group *E*	**3.33±0.67**	3.22±0.46	3.11±0.27
	Group *NE*	**3.88±0.48**	3.65±0.31	3.33±0.45
Q4	Group *E*	**4.72±0.33**	4.61±0.44	4.28±0.57
	Group *NE*	**3.88±0.32**	3.52±0.25	3.43±0.3

the answers to Q1 show that users were satisfied with the completeness of our summary. Q2 and Q3 also show the user's satisfaction, although group *E*'s rating is slightly lower than group *NE*'s. This is probably because the experienced participants wanted to see all interesting highlights in the summary, but some were missing. The inexperienced participants did not have such a firm predilection. In Q4, group *NE* found the summaries more redundant than group *E*, in a way that group *NE* preferred watching also non-active segments before the action starts for a better understanding of the context.

When comparing summary types, it can be observed that the clustering-based baseline has the lowest scores for all the questions. Overall, group *E* rated the summaries created with their annotations higher, except in Q4. For group *NE*, the difference between summaries generated with their annotations or with group *E*'s is not noticeable. Regarding length, 30 second summaries obtained the best evaluation for all questions and user groups. We consider the reason is that 20 second summaries contained some incomplete highlights that were filled in the 30 second ones, but in the 40 second summary, newly added highlights were incomplete. The summary for video (a) was ranked higher for all questions and both groups, which is coherent since it has the highest f-score.

Some participants in group *NE* commented the usefulness of our method to

extract highlights based on actions, and the time they can save by watching the summary instead of the whole video. They stressed the importance of context to understand what is happening in some of the highlights. Group E stated that the reason they lowered the score of the videos is that in Kendo it is important to observe the actions after hitting the opponent as well (even if they are not interesting) in order to decide if it was a good hit. In many highlights, this part was not extracted. However, when creating a summary for a given length, our method gives priority to extracting new interesting highlights rather than adding less interesting sub-sequences to the existing ones. All our participants preferred watching longer highlights rather than a larger number of them.

4.4 Summary

In this chapter we have presented a novel method for generating video summaries with highlights of user-generated sports video by using HAR, which is used to train a highlights model based on viewers' opinion on which sections of the original video were interesting. Our experiments and the positive responses from the survey showed that our method was able to successfully extract highlights using HAR, despite our HAR was not perfect. We believe the reason is that our method does not directly rely on HAR results, but on its intermediate outputs, which can leverage the ambiguity among different action classes. Although we experimented with only one type of sport, *i.e.*, Kendo, our method is applicable to other similar sports. The contributions of this work are summarized as follows:

- We proposed a novel method for summarizing user-generated sports video based on HAR from a self-recorded RGB-D video sequence. To the best of our knowledge, this is the first attempt to use this kind of analysis for video summarization. Our method is suitable for sports that can be recorded at a close distance.

- We evaluated the performance of our method both objectively and subjectively to show its effectiveness and accuracy. We carried out a survey of users with and without experience in the sport to investigate the adequacy of our method to their particular preferences.

5 Summarization of user-generated sports video using deep action features

5.1 Overview

User-generated sports video (UGSV) summarization basically inherits the intricacies of user-generated video summarization. In the absence of editing conventions, extracting high-level semantics is not trivial. However, given that the target is a sport, we can leverage this domain knowledge to facilitate the extraction of high-level semantics. As introduced in Chap. 4, our idea towards this direction is to utilize players' actions, which are the main constituents of a game. Our previous work in Chap. 4 applies an action recognition technique to sports video to find combinations of actions that interest viewers using a hidden Markov model with Gaussian Mixture emissions.

To the best of our knowledge, our previous work in Chap. 4 is the only one that tries UGSV summarization based on players' actions. One major drawback of this work is that it takes a classic approach: it uses handcrafted features for action recognition and a conventional classifier. The recent trend of deep neural networks has demonstrated the power of feature learning, in which a neural network is trained in an end-to-end manner from its input to the top layers or at least partially from one of its layers to the top. Another interesting direction to extend our previous work is the use of different features. In Chap. 4, we only use body joint positions as a cue for action recognition. They provide a rich information on players' action, but miss other potential cues for summarization in the appearance of the scene. At least, appearance is useful when the joint

position estimation (e.g. [121]) fails.

In this chapter, we extend our previous method in Chap. 4 by employing a two-streams deep neural network [29, 91]. Our new method takes two different types of inputs, i.e., RGB frames of video as well as body joint positions, each of which is transformed through two separated neural networks (i.e., streams). These two streams are then fused to form a single action representation for finding the highlights. In the previous method, we separately train an action recognizer for the target sport. In contrast, our new method no longer needs such an action recognizer; our network is trained from a lower to the top layers using an extended UGSV summarization dataset, which is three times bigger than the one presented in 4.3.1.

Given our methodology, our target sports should meet the following conditions: (1) a game consists of a series of recognizable actions performed by each player and (2) players can be recorded from a close distance for joint position estimation. We, however, believe that the idea of using action recognition-related features for UGSV summarization is still valid for most types of sports.

5.2 Deep neural network for UGSV summarization using two motion streams

In this work, we formulate UGSV summarization as a problem of classifying a video segment in the original video into interesting (and thus included in the summary) or uninteresting. We design a two-stream neural network for this problem and train it in a supervised manner with ground truth labels provided by multiple annotators.

Figure 5.1 shows an overview of our method. It first divides the input video into video segments $S = \{s_t\}$, in which RGB frames may be accompanied by their corresponding depth maps. A video segment s_t is then fed into our two-stream network. The body joint-based feature stream takes RGB frames (and depth maps) in s_t to obtain the body joint-based features x_t, and the holistic feature stream computes holistic features y_t from RGB frames. The former stream captures the players' motion in detail by estimating their body joint positions explicitly. The latter is to represent the entire frames in the video segment,

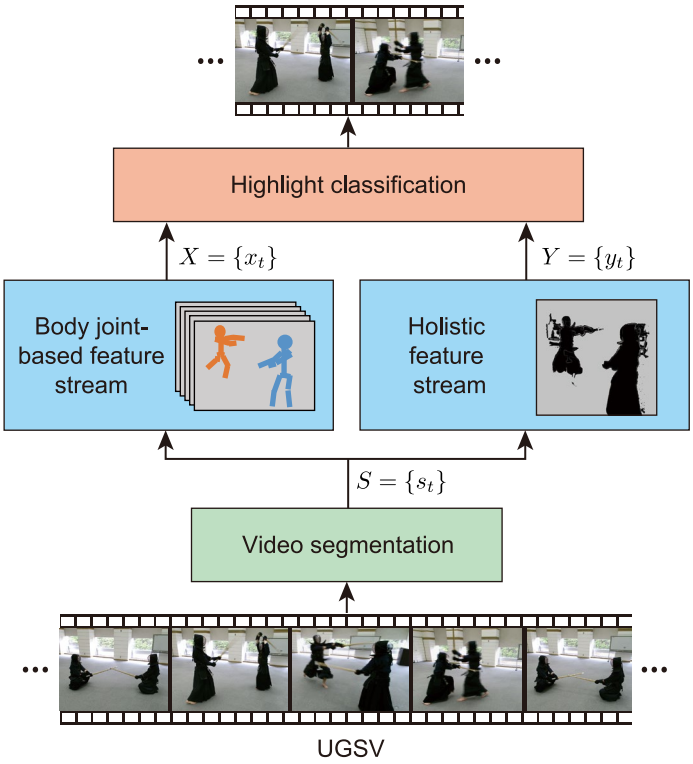


Figure 5.1: An overview of our method for generating a summary of UGSV based on the players' actions. We use two types of features to represent players' actions, i.e. body joint-based and holistic, for extracting highlights from the original video.

which can be helpful to encode, e.g., the relationship between the players. Our features $X = \{x_t\}$ and $Y = \{y_t\}$ are then fed to the highlight classification block to find the highlights of the input video. This block takes into account the temporal dependencies among the video segments. Our highlight summaries are a concatenation of the segments classified as interesting.

In both the body joint-based feature extraction and the highlight classification blocks we use long short-term memory (LSTM) cells for modeling the temporal relationship of our features. LSTM has been previously used for video summarization [116, 119], and action recognition with both hand-crafted features [56] and deep features from CNN [69, 101]. First, we introduce this type of RNN.

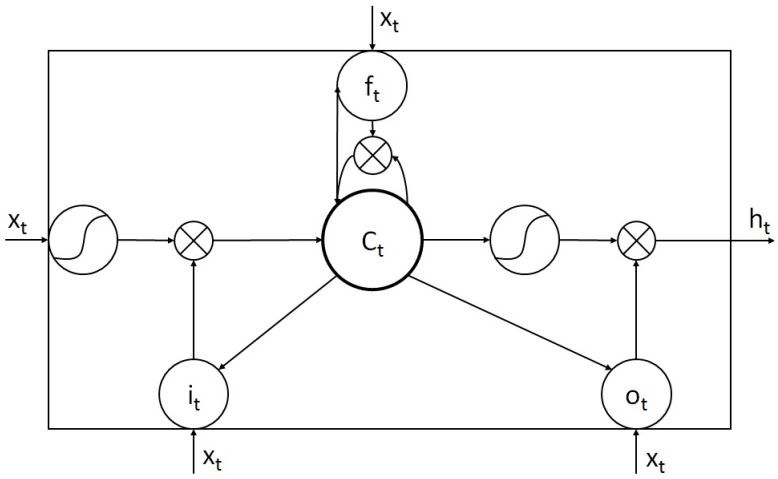


Figure 5.2: Architecture of a long short-term memory cell, obtained partially from [33].

5.2.1 Long short-term memory

Long short term memory (LSTM) [37] is a type of recurrent neural network used for modeling temporal sequences. An LSTM network takes the input sequence $X = (x_1, x_2, \cdots, x_T), x_t \in R^d, t \in [1, T]$ and calculates a hidden vector sequence $H = (h_1, h_2, \cdots, h_T), h_t \in R^{d'}, d' < d$ such that it outputs a reconstructed sequence $Y = (y_1, y_2, \cdots, y_T)$. Unlike other recurrent networks, LSTM are more effective at finding and modeling long-range context along a time-series, and they have been previously used in video classification tasks [69]. Figure 5.2 shows the typical operation of an LSTM cell, which uses learning gate functions to determine whether an input is significant enough to remember or it should be forgotten, and when it should be sent to the output.

The following equations describe how a layer of LSTM memory cells is updated at every time-step t. The terminology used is:

- x_t is the input to the memory cell at time t.
- i, f and o are the input, forget and output gates respectively.
- A and h have the same size, and denote the cell activation states and hidden states respectively.

- W, U and V are weight matrices for the input x, hidden state h, and cell activation a respectively. For example, the W_f matrix represents the connections between the input and the forget gate.
- b_i, b_f, b_a and b_o are bias vectors.
- σ is the logistic sigmoid function.

First, we compute the values for i_t (input gate) and $\widetilde{A}_t$ (candidate value for the states at time t):

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5.1}$$

$$\widetilde{A}_t = tanh(W_a x_t + U_a h_{t-1} + b_f) \tag{5.2}$$

Second, we compute the value f_t (activation of the forget gate at time t):

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5.3}$$

Then, we calculate A_t (new state at time t):

$$A_t = i_t * \widetilde{A}_t + f_t * A_{t-1} \tag{5.4}$$

With the new state, we can calculate the value of the memory cell's output gate and then the output hidden state:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o A_t + b_o) \tag{5.5}$$

$$h_t = o_t * tanh(A_t) \tag{5.6}$$

5.2.2 Video segmentation

Various methods have been proposed to segment a video based on, e.g., its content [12], but in our method we uniformly segment the original input video into multiple segments s_t, i.e., $S = \{s_t | t = 1, \ldots, T\}$, where T is the number of the video segments in S and s_t is the video segment that contains frames from $t-1$ to $t+\tau-1$ second as shown in Figure 5.3. Since most actions last only a very short

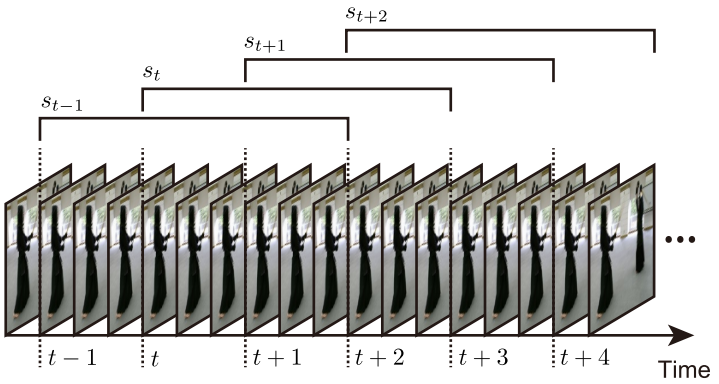


Figure 5.3: Video segmentation. Video segment s_t contains frames in-between $t-1$ and $t+2$ seconds. Each video segment overlaps with adjacent ones for two seconds.

period of time, we need short video segments for a finer labeling of highlights. We choose $\tau = 3$ seconds, so adjacent video segments overlap with each other by 2 seconds. Each segment s_t contains a different number of frames, especially when the input video is captured with an RGB-D camera (e.g., Microsoft Kinect), due to automatic exposure control.

5.2.3 Body joint-based feature stream

For this stream (Figure 5.4), in order to obtain a detailed representation of players' actions, we use a sequence of position of the players' body joints (e.g., head, elbow, etc.) that represent the movement of the players regardless of their appearance. In this work, we employ two types of joint representations, i.e., 3D positions from depth maps or 2D positions from RGB frames.

In the case of 3D body joint positions, we use a skeleton tracker (e.g., [120]) as in Chap. 4, which estimates the 3D positions from depth maps. The 3D positions are usually in the camera coordinate system, so they are view-dependent, which introduces extra variations. Therefore, we transform the 3D positions from the camera coordinate system to each player's coordinate system, whose origin is at one of the body joints (e.g. torso).

In the absence of depth maps, which is likely in current user-generated video, we can still estimate 2D body joint positions from RGB frames. Recent methods

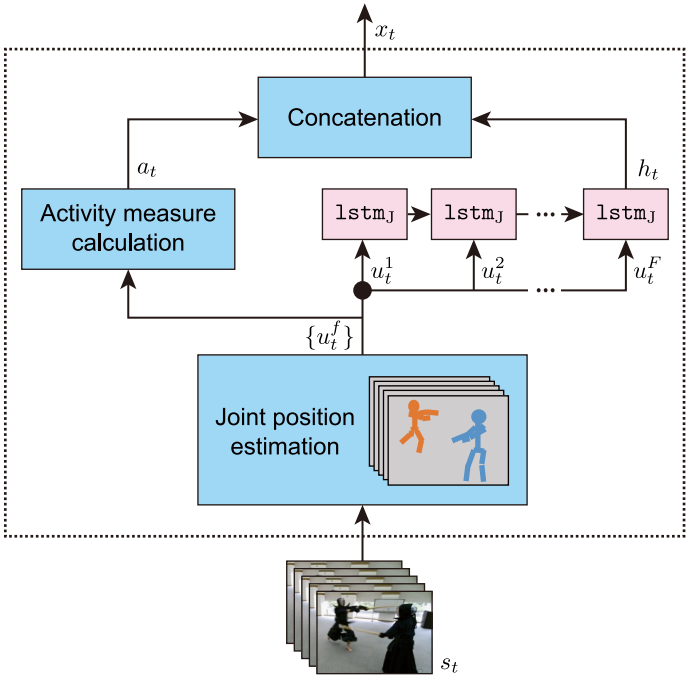


Figure 5.4: We feed an LSTM with the body joint positions estimated from players on each frame x_t^f to model their temporal dependencies and extract a feature vector h_t. We also use these body joint positions to calculate an activity measure for all players a_t. Our body joint-based feature vector is the concatenation x_t.

in human pose estimation leverage 2D CNNs to learn the spatial relationships among human body parts and estimate the 2D joint positions [112]. Such 2D positions are not as robust against view variations as 3D positions, but they can be extracted from RGB frames alone without using depth maps. Given the 2D body joint positions, we also transform them to positions relative to the player's coordinate system to make them translation invariant.

The use of an activity measure works positively when extracting highlights (Chap. 4). To calculate the activity measure a, we divide the volume (or plane for the 2D case) around a player into regions and calculate the ratio r_v of the number of frames in the video segment in which the joint j falls into region v. The activity measure a is defined as the entropy obtained based on r_v. For each

joint j, we compute

$$e_j = -\sum_v r_v \log(r_v). \tag{5.7}$$

The activity measure is calculated by

$$a = \sum_{j=1}^{J} e_j. \tag{5.8}$$

We calculate the activity measure for each player in a segment. More details on the activity measure can be found in Chap. 4.

We represent joint j of player q in frame f using 3D or 2D relative body joint positions u_{qj}^f in $\mathbb{R}^3$ or $\mathbb{R}^2$ (a row vector). Then, $u_t^f = (u_{11}^f \cdots u_{QJ}^f)_t$ is the concatenation of the body joints of all players in frame f for the video segment s_t, where Q and J are the numbers of players and joints. As shown in Figure 5.4 we pass vectors u_t^1 to u_t^F through an LSTM to model the temporal dependencies of the players' body joint positions in s_t. After feeding the last vector x_t^F, we take the hidden state vector h_t of the LSTM as a representation of $\{u_t^f\}$. We reset the state of the LSTM to all zeros before feeding the next video segment. We presume that the number of players Q does not change. However, some players can be out of the field-of-view of the camera. In that case, we pad the corresponding elements in u_t with zeros.

Our method represents a video segment s_t by concatenating the LSTM output and the activity measure of all players in one vector

$$x_t = (h_t \;\; a_t), \tag{5.9}$$

where a_t is the concatenation of $(a_{t1} \;\; \cdots \;\; a_{tQ})$ and a_{tq} is the activity measure of player q in s_t.

5.2.4 Holistic feature stream

This stream encodes a video segment s_t in a spatio-temporal representation. We rely on state-of-the-art three-dimensional convolutional neural networks (3D CNN) over RGB frames. While in 2D CNN convolution and pooling operations are performed only spatially, in 3D CNN are done also temporally. Figure 5.5

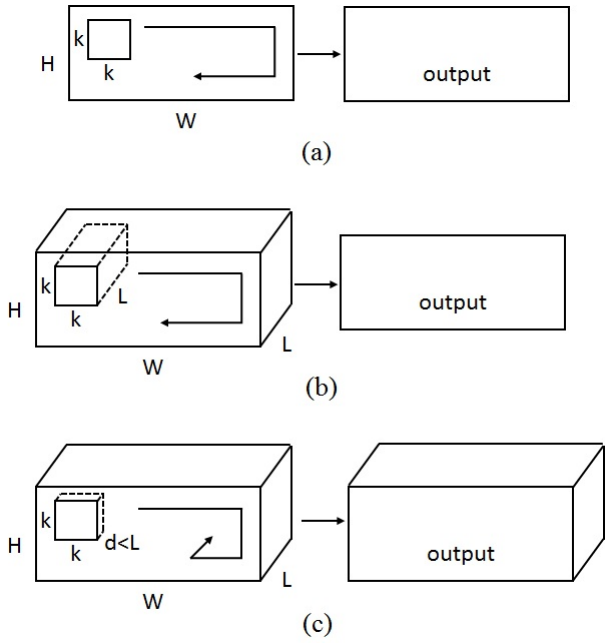


Figure 5.5: A 2D convolution on an image results in an image (a). A 2D convolution on a video volume results in an image (b). A 3D convolution on a video volume results in a volume, that is, the temporal information is preserved (c).

(obtained from [101]) shows the difference. A 2D convolution on an image will produce an image; a 2D convolution on multiple images will also produce an image. This means that 2D CNN lose temporal information on each convolution, and thus they are not particularly sensitive to temporal modeling. On the other hand, a 3D convolution on multiple images results in a volume, preserving temporal information of the input signal.

This stream encodes a video segment s_t in a spatio-temporal representation. We rely on state-of-the-art 3D CNN over RGB frames. Training a 3D CNN from scratch requires thousands of videos [41], which are not available for our task. Recent work on deep neural networks for computer vision [29, 101, 118] show that the activations of an upper layer of a CNN can be used for other related tasks without requiring fine-tuning. Thus, we can instead use 3D CNN whose parameters are pre-trained with large-scale datasets to leverage a huge amount of labeled training data [39]. For example, since we consider that our

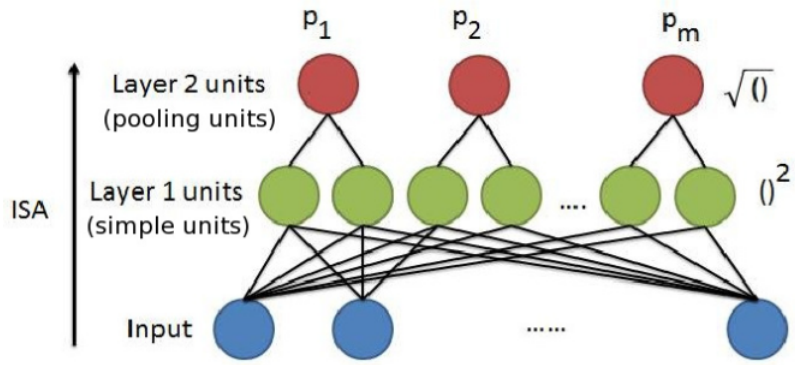


Figure 5.6: Independent subspace analysis network architecture with a subspace size of 2: each pooling unit looks at 2 simple units (obtained from [45])

UGSV summarization task is related to action recognition, we can use a publicly available dataset for action recognition, such as Sports-1M [41].

Since we hypothesized that players' actions allow modeling the highlights of the video, we consider two different 3D CNN for feature extraction that are successful in capturing motion information in videos: Independent subspace analysis-based CNN (CNN-ISA) [45] and Convolutional 3D (C3D) [101].

On the one hand, CNN-ISA learns features that are robust to local translation while being selective to frequency, rotation and velocity. This allows discarding background information and camera motion to focus on the actions performed by the actors in the video. Figure 5.6 and 5.7 shows the architecture of the network. It uses a representation based on spatiotemporal cuboids that describe the local spatiotemporal video patch, that is flattened into a vector of input features. The learned features are then convolved with a larger region of the input data, and the outputs of this convolution step are inputs to the layer above. Finally, learning is carried out by updating the network parameters with batch projected gradient descent. The size of the input video blocks, ISA equations and other details of the model can be found in the original paper [45]. CNN-ISA achieves state of the art precision in well-known datasets for action recognition such as YouTube [51], Hollywood2 [57] and UCF sports [82].

On the other hand, C3D learns features by focusing on appearance in the first few frames and tracking the salient motion in the subsequent frames selectively.

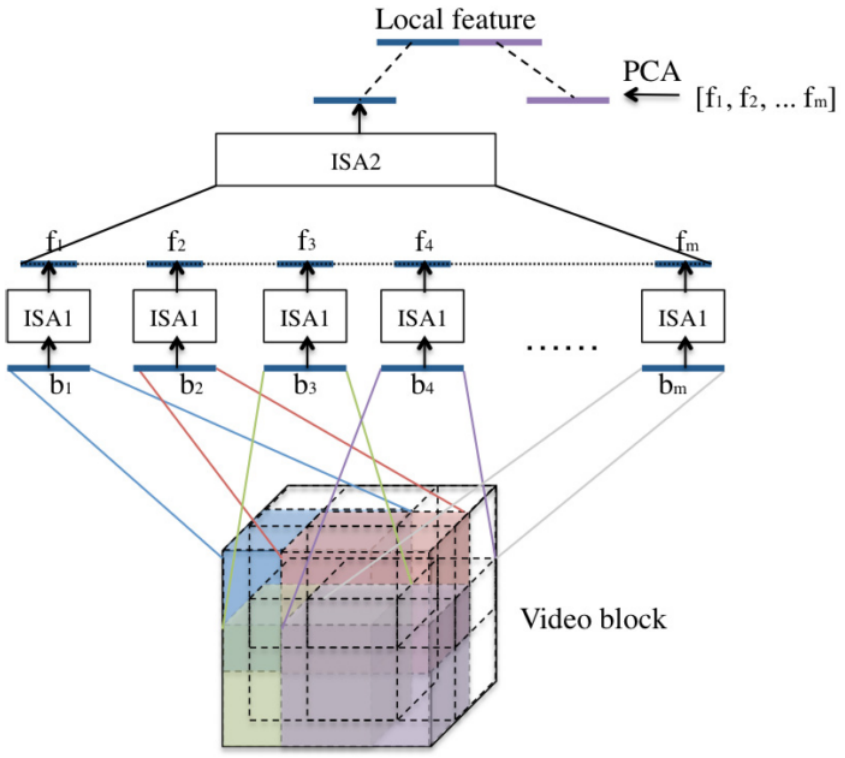


Figure 5.7: In the CNN-ISA, the ISA network in the second later is trained on the combined activations of the first layer (obtained from [45])

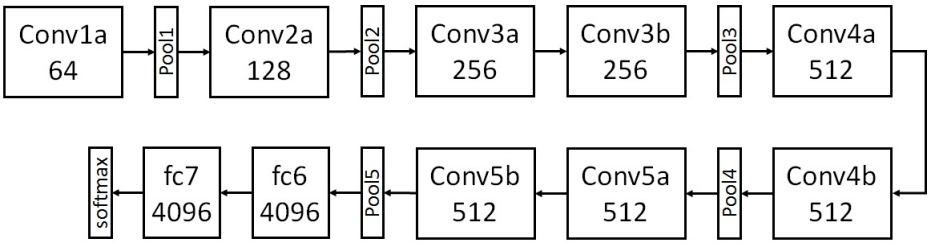


Figure 5.8: Architecture of the C3D network (obtained from [101])

The architecture of this network is depicted in Figure 5.8; it has 8 convolution layers, 5 pooling layers, followed by 2 fully connected layers and a softmax output layer. The 3D convolution filters are $3 \times 3 \times 3$ with stride $1 \times 1 \times 1$. The 3D pooling layers are $2 \times 2 \times 2$ with stride $2 \times 2 \times 2$ except for `pool1` which has a kernel size of $1 \times 2 \times 2$ and stride $1 \times 2 \times 2$ with the intention of preserving the

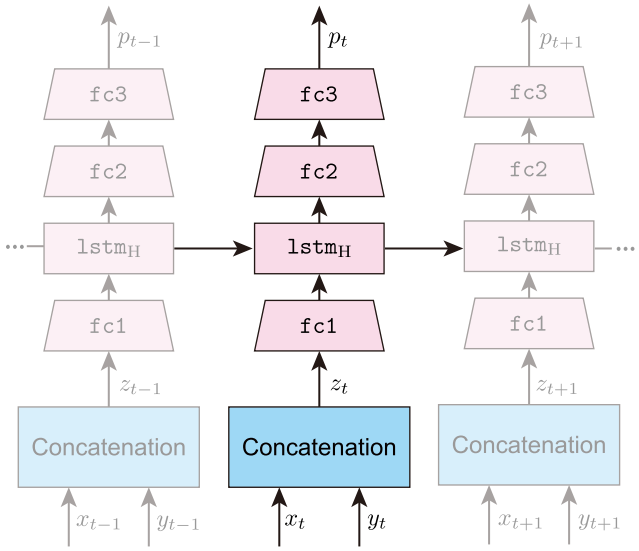


Figure 5.9: Neural network architecture for highlight classification, which consists of a single LSTM layer and several fully-connected layers. We feed the body joint-based features x_t and holistic features y_t extracted from video segment s_t to calculate its probability p_t of being interesting.

temporal information in the early phase. These and more details can be found in [101]. C3D pre-trained with the Sports-1M dataset achieves state of the art precision for action recognition with the UCF101 dataset [94].

This stream represents video segment s_t using a holistic feature vector y_t, which is the output of one of the aforementioned 3D CNNs.

5.2.5 Highlight classification using LSTM

Figure 5.9 shows the network architecture designed to model highlights of UGSV using our features x_t and y_t. We again use an LSTM in order to model the temporal dependencies among video segments, and the network outputs the probability p_t that the video segment s_t is interesting. We first concatenate the features to form vector $z_t = (x_t \;\; y_t)$. Vector z_t then goes through a fully-connected layer to reduce its dimensionality.

We consider that video segments are temporally related to each other; e.g., a skillful boxer first feints a punch before hitting to generate an opening in the

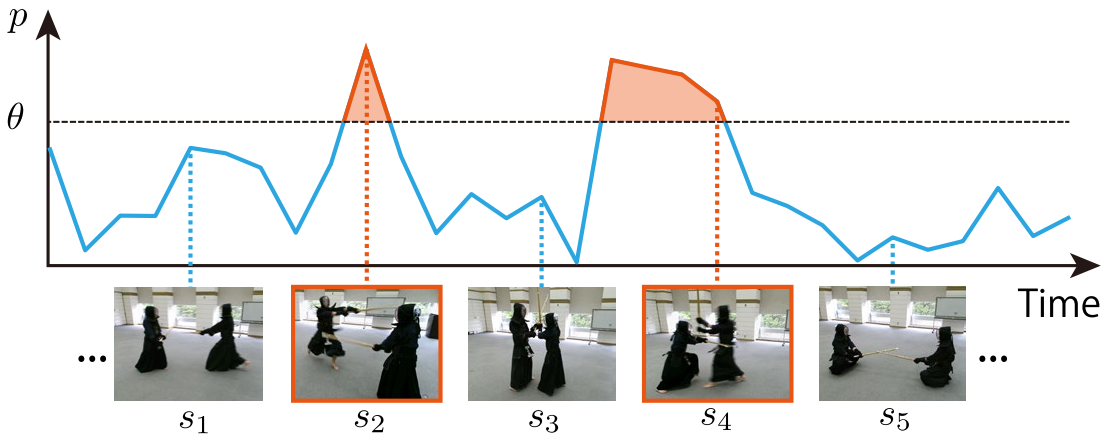


Figure 5.10: We generate a summary by concatenating segments whose probability p_t of being highlight surpasses a certain threshold θ. The threshold is chosen to fit the summary length.

defense. Existing work in video summarization uses LSTMs to extract video highlights [116], since it allows to model temporal dependencies across longer time periods than other methods [69]. We follow this idea and introduce a LSTM layer to our network. The hidden state of the LSTM from each time step goes through two fully-connected layers, resulting in a final softmax activation of two units, which correspond to "interesting" and "uninteresting."

Our method provides the control over the length L of the output summary. Therefore, instead of hard decision, we deem the softmax activation of the unit corresponding to "interesting" as the probability p_t of the segment s_t being interesting and apply skimming curve formulation [102] to the sequence of probabilities by decreasing the threshold θ from 1 until it finds a set of segments whose total length is largest below L as shown in Figure 5.10. The segments whose probability exceeds θ are concatenated to generate the output summary in the temporal order. In this way, a resulting summary may contain multiple consecutive interesting segments.

5.2.6 Network training

We use pre-trained CNN in the holistic features stream (i.e. CNN-ISA or C3D), whereas we train our LSTMs and fully-connected layers from scratch. That is,

during training, the parameters in the holistic feature stream are fixed, and those in the body joint-based feature stream (i.e., $\texttt{lstm}_J$) and highlight classification (i.e., `fc1`, $\texttt{lstm}_H$, `fc2`, and `fc3`) are updated.

Our UGSV dataset contains video and ground truth labels $l_t \in \{0, 1\}$ for every 1 second, where $l_t = 1$ means that the period from t to $t + 1$ second of the video is "interesting" and $l_t = 0$ otherwise. We assign label l_t to s_t, which covers the frames in $t - 1$ to $t + 2$ since s_t captures the period from t to t_1 second in its center.

For training, we used cross-entropy loss ℓ defined as

$$\ell = \sum l_t \log p_t. \tag{5.10}$$

5.3 Experiments

We evaluate our method objectively and subjectively. For the objective evaluation, we compare the performance of our method when using different representation of the players' actions. More concretely, we evaluate body joint features only (3D or 2D), holistic motion features only (CNN-ISA or C3D), and the combination of both. Then, we study the completeness of the highlights of the generated summaries. For the subjective evaluation, we surveyed users with and without experience in the sport to study their opinion about our summaries.

5.3.1 Implementation details

For evaluation, we chose Kendo (Japanese fencing) as an example sport, which is a martial art featuring two players and a set of recognizable actions (e.g., attacking and parrying). We extended the UGSV Kendo dataset used in Chap. 4, which contains 90 minutes of self-recorded Kendo matches divided in 10 RGB-D videos taken with a Microsoft Kinect v2, by adding 18 more self-recorded RGB-D Kendo videos. The total length of our videos is 246 minutes, with a framerate of around 20 fps.

Our body joint-based feature stream was configured for $Q = 2$ players, since Kendo is a two-player sport. We used the tracker in [120] for estimating $J = 15$ 3D body joint positions from depth maps, more specifically: *head*, *neck*, *torso*,

right shoulder, *right elbow*, *right wrist*, *left shoulder*, *left elbow*, *left wrist*, *right hip*, *right knee*, *right ankle*, *left hip*, *left knee* and *left ankle*. For estimating the 2D positions of the players' joints from the RGB frames, we use the CNN-based method proposed by Linna et al. [50]. We initialized the network's parameters with Linna et al.'s human pose estimation dataset and fine-tuned it with our extended UGSV Kendo videos. This network provides $J = 13$ joints (same as the 3D case except *neck* and *torso*). Therefore, the size of vector u_t^f is $Q \times J \times 3 = 90$ in the case of 3D positions and $Q \times J \times 2 = 52$ in the case of 2D. Since we made the size of $\texttt{lstm}_J$ the same as the input, and the size of a_t is $Q = 2$, the feature vector for this stream is $x_t \in \mathbb{R}^{92}$ for 3D, or $x_t \in \mathbb{R}^{54}$ for 2D.

For the holistic feature stream, we used either the CNN-ISA [45] or C3D [101] network. Since our UGSV Kendo dataset is not big enough to train these CNNs from scratch, we used networks pre-trained with an action recognition dataset. CNN-ISA was trained in an unsupervised way with the Hollywood2 dataset consisting of 2859 videos [57]. For this network, we followed the configuration in [106] and used a vector quantization representation of the extracted features with a codebook size of 400, resulting in a feature vector $y_t \in \mathbb{R}^{400}$ for each segment s_t. C3D was trained with the Sports-1M dataset [41], which consists of 1.1 million videos of sports activities. We extracted C3D features as indicated in [101] by uniformly sub-sampling 16 frames out of around 60 frames in s_t (the number of frames in s_t may vary for different segments due to the variable framerate of Microsoft Kinect v2) and then extracting the activations from layer `fc6` (i.e., $y_t \in \mathbb{R}^{4096}$).

Our method was implemented in Chainer [99]. The learning rate is calculated by the adaptive moment estimation algorithm (Adam) [42] with $\alpha = 0.001$. We introduced sigmoid activation after our fully-connected layers. Tables 5.1, 5.2 and 5.3 summarize the number of learnable parameters for each layer, which varies depending on the choice of features.

5.3.2 Results

We invited 15 participants to our experiment and divided them in two groups, experienced (*E*, 5 people) and inexperienced (*NE*, 10 people), according to their experience in the target sport (i.e., Kendo). We considered that the highlights

Table 5.1: Size of the learnable parameters in our network ($input \times output$) when using only body joint-based features. Feature vector sizes are detailed in Section 5.3.1)

	Body joint-based features only		
	3D joints	2D joints	Action recognition
$\texttt{lstm}_J$	90×90	52×52	—
fc1	92×50	54×50	402×400
$\texttt{lstm}_H$	50×50	50×50	400×400
fc2	50×20	50×20	400×100
fc3	20×2	20×2	100×2

Table 5.2: Size of the learnable parameters in our network ($input \times output$) when using only holistic features. Feature vector sizes are detailed in Section 5.3.1)

	Holistic features only	
	CNN-ISA	C3D
$\texttt{lstm}_J$	—	—
fc1	400×400	4096×400
$\texttt{lstm}_H$	400×400	400×400
fc2	400×100	400×100
fc3	100×2	100×2

that *E* and *NE* groups prefer would vary greatly from each other, and we wanted to evaluate how well our method adapts to their needs. Then, we asked them to manually annotate the highlights of our 28 videos. We obtained the ground truth labels of our videos for both *E* and *NE* groups separately, considering that each one-second period in video is interesting if 40% of the participants agreed. Due to group *E*'s technical knowledge of Kendo, their highlights contain very specific actions (e.g., decisive strikes, counterattacks). On the other hand, group *NE* selected not only strikes but also more general actions (e.g., parries, feints), so their labeled highlights are almost three times as long as group *E*'s (see the

Table 5.3: Size of the learnable elements of our network (*input* × *output*) when using both body joint-based and holistic features. Feature vector sizes are detailed in Section 5.3.1)

	Body joint-based and holistic features	
	3D joints + CNN-ISA	2D joints + CNN-ISA
$\texttt{lstm}_J$	90×90	52×52
`fc1`	492×400	454×400
$\texttt{lstm}_H$	400×400	400×400
`fc2`	400×100	400×100
`fc3`	100×2	100×2

Figure 5.11: Sample segments in a Kendo match that our method classified as highlights when generating a summary.

durations in the Appendix).

We trained our network separately with each group's ground truth labels in the leave-one-out (LOO) fashion, i.e., we used 27 videos for training and generated a summary of the remaining one for evaluation. The CNN for 2D pose estimation was trained independently before each experiment, fine-tuning it with the same 27 videos and estimating the joints of the video used for evaluation. Repeating this process for each video results in 28 experienced summaries and 28 inexperienced summaries. We generated summaries with the same length as the ground truth. Figure 5.11 illustrates some example frames of a video as well as highlight frames extracted by our method (framed in red).

Objective evaluation by segment f-score

We evaluate the ability of our method to extract highlights in terms of the f-score. In our method, one-second period of video is:

- true positive (TP) if in the summary and $l_t = 1$,
- false positive (FP) if in the summary but $l_t = 0$,
- false negative (FN) if not in the summary but $l_t = 1$, or
- true negative (TN) if not in the summary and $l_t = 0$.

The f-score is then defined as

$$\text{f-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{5.11}$$

Tables 5.4 and 5.5 show the f-scores for the summaries generated with the labels of both *E* and *NE* groups. Firstly, Table 5.4 compares the performance of the feature combinations described in Section 5.3.1. The upper part of the table presents the results of using body joint-based features only (with activity measure). The second part presents the results of using holistic features only. The third part shows the results of using the features from our previous work in UGSV summarization (Chap. 4). We obtained the features by feeding a 3D body joint representation of players' actions to the action recognition method in Chap. 3, and taking the action classification results. Lastly, the lower part shows the results obtained from the combination of body joint-based and holistic features.

Then, table 5.5 compares our proposed architecture with our previous method (Chap. 4), which uses a Hidden Markov Model with Gaussian mixture emission (GMM-HMM) over the action recognition results mentioned in the previous paragraph, and k-means clustering. Such clustering-based method is widely accepted as a baseline for user-generated video summarization [20]. For our k-means clustering baseline, we cluster our video segments S based on the concatenation *3D joints and CNN-ISA* features and take each cluster centroid. We configured the number of clusters for each video so that the resulting summary length is equal to the ground truth's.

Table 5.4: F-score comparison of different combinations of features in our method.

Features	Group *E*	Group *NE*
3D joints	0.53	0.83
2D joints	0.45	0.77
CNN-ISA	0.5	0.79
C3D	0.27	0.60
Action recognition (Chap. 4)	0.48	0.76
3D joints + CNN-ISA	**0.58**	**0.85**
2D joints + CNN-ISA	0.57	0.81

Table 5.5: F-score comparison of out method (3D joints + CNN-ISA) with other UGSV summarization methods.

Method	Group *E*	Group *NE*
Our method	**0.58**	**0.85**
GMM-HMM (Chap. 4)	0.44	0.79
k-means clustering	0.28	0.61

When using a single feature (i.e. 3D joins, 2D joints, CNN-ISA, C3D, or action recognition), 3D joints obtain the best performance. Although C3D features perform well in action recognition tasks [101], its results were significantly worse than the other features in our summarization task. Since the dimensionality of the C3D features is prominently large compared to others, we might not have been able to train the network well with our dataset. Fine-tuning C3D over our dataset might improve its performance. On the other hand, CNN-ISA, which also uses RGB frames, obtains better results than C3D and even 2D joints. This implies that we can also obtain from RGB frames features that allow us to model UGSV highlights. The drop in performance found between 3D joints and 2D joints may indicate that view variations in the same pose affects negatively our body joint-based features stream. The action recognition feature had an intermediate performance. One of the reason can be that the action recognition feature is

based on a classic approach for classification and some useful cues in 3D body joint positions degenerated in this process. From this result, the features that performed better for highlight classification are CNN-ISA holistic features and 3D body joint-based features.

Several state-of-the-art methods in action recognition tasks enjoy a boost in performance by combining handcrafted spatio-temporal features (e.g., dense trajectories) and those learned via CNNs [29,101]. This is also true in our case, where the combination of CNN-ISA with 3D joints achieves the best performance. The combination of CNN-ISA with 2D joints also provides a considerable boost in performance, especially for the experienced summaries. This confirms our hypothesis that a two-streams architecture also provides better results for UGSV summarization.

Finally, as shown in Table 5.5, our method outperformed both the previous work and the clustering-based baseline. While clustering allows to show a wider variety of scenes in the summary, this is not a good strategy for UGSV summarization, which follows a different criterion on interestingness. Our proposed method also outperforms the previous work, that used the classification results of an action recognition method and fed them to a GMMHMM for highlight modeling.

Thus, our method outperforms both the highlights model trained on action recognition results and also the feature representation based on action recognition results (Table 5.4). We can conclude that it is not necessary to explicitly recognize the players' actions for UGSV summarization; it might actually degrade the performance compared to directly using action recognition features.

Objective evaluation by highlight completeness

A highlight may consist of consecutive video segments. This means that, while missing one segment may not have much impact on the f-score, it affects the continuity of the video, and thus, the comprehensibility and the user experience of the summary. For this, we define a criterion to evaluate the completeness c of an extracted highlights as the fraction of overlap between the extracted highlight and its associated ground truth highlights (Figure 5.12). Associating extracted and ground truth highlights is not trivial, and we did this using a greedy algorithm,

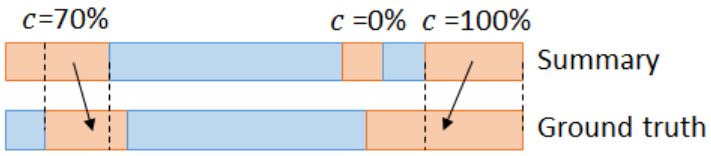


Figure 5.12: Association of highlights by greedy algorithm. Each highlight in the ground truth is uniquely associated to a highlight in the generated summary (two summary highlights cannot share the same ground truth highlight). The completeness of a summary highlight is the percentage of overlap with the ground truth (0% if unassociated).

in which the total c of all highlights is maximized. We deemed an extracted highlight is a TP if its completeness c is greater than a certain percentage C%, and according to this we calculated precision and recall of our highlights as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5.12}$$

In this experiment, we moved the threshold θ from 0 to 1 over the probability p_t to generate the recall-precision curve of group E and NE.

Figure 5.13 shows the curves produced for $C = 50\%$, 70%, and 90%. We observe that reducing C to 50% increases the number of complete highlights significantly. We attribute the presence of incomplete highlights to our highlight extraction; first the *high p segments* are extracted, and then the highlight is completed with *low p segments* as the threshold θ decreases (Figure 5.14). But before a highlight is completed, *high p segments* from other highlights are extracted and, in some cases, the *low p segments* are never extracted. In particular, the parts before and after an interesting technique normally correspond to *low p segments*, since they are not present in every ground truth highlight annotated by our participants.

Also, the reason there are more incomplete segments (less TP) in the NE summaries is that the inexperienced group annotated a larger number of highlights.

Subjective evaluation

We asked the same participants who annotated the original videos to participate in a survey, in order to assess their opinion on the ground truth and summaries

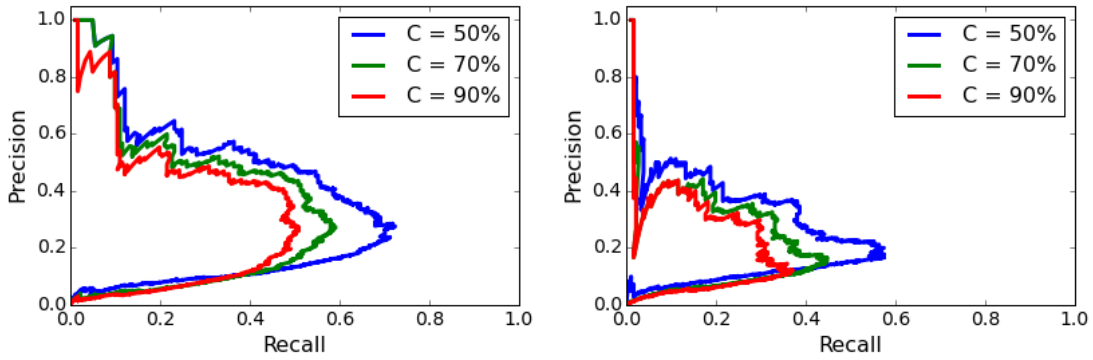


Figure 5.13: Recall-precision curves for different completeness values (up: labels *E*, down: labels *NE*). The gap between the completeness $C = 50\%$ and $C = 70\%$ shows that a significant number of our highlights are missing at most half of the interesting segments.

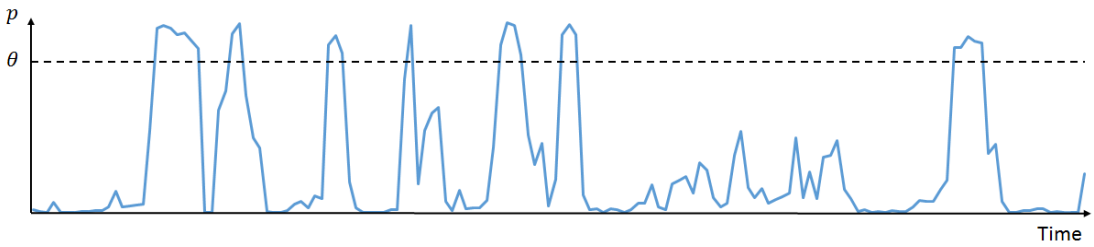


Figure 5.14: Recall-precision curves for different completeness values (left: labels *E*, right: labels *NE*). The gap between the completeness $C = 50\%$ and $C = 70\%$ shows that a significant number of our highlights are missing at most half of the interesting segments.

we generated. We chose the three videos with the highest, median and lowest f-scores (averaged over groups *E* and *NE*). For each video, we showed participants our ground truth and the summaries generated with the best feature combination (i.e., *3D joints + CNN-ISA*) using both group *E* and *NE* labels. As a result, each participant watched 12 videos (3 f-scores × 4 video types).

We asked the participants to (Q1) assign a score in a Likert scale from 1 (very few highlights are interesting) to 5 (most highlights are interesting) according to their satisfaction with the contents of each of the 12 videos. We also asked them to (Q2) state their opinion on the videos and the criteria they followed when

assigning a score. Tables 5.6 and 5.7 show the results of Q1 grouped by video type and video f-score. The scores are averaged for group *E* and *NE* separately.

Table 5.6: Subjective evaluation results according to the video type. Each cell consists of the mean ± standard deviation of the survey scores.

	Video type			
	Ground truth *E*	Ground truth *NE*	Summary *E*	Summary *NE*
Group *E*	3.2±0.99	3.07±1.04	2.6±1.23	2.73±0.87
Group *NE*	3.57±0.72	3.5±1.07	3.2±0.83	2.9±0.97

Table 5.7: Subjective evaluation results according to the video f-score. Each cell consists of the mean ± standard deviation of the survey scores.

	Video f-score		
	Highest	Median	Lowest
Group *E*	3.3±0.95	2.85±0.97	2.55±1.18
Group *NE*	3.48±0.83	3.03±0.91	3.38±0.95

Regarding Q1, in terms of the video type, both experienced and inexperienced participants assigned a higher score to the ground truth videos than the summaries, since some summaries contain uninteresting video segments and the completeness of the highlights is worse. The reason why the ground truth did not get a perfect score is mainly due to two factors: (1) The ground truth summaries are created by combining labels from several participants via majority voting, so the original labels of each participant is lost. (2) The ground truth also contains incomplete highlights due to errors when the participants annotated the videos. Also, experienced participants preferred the *NE* ground truth than the *E* summaries; this may be because they do not find our extracted highlights interesting when the context is missing. On the other hand, the inexperienced participants tend to appreciate more the highlights from the experienced participants' than their own. We believe this is because they are briefer and contain

certain techniques (e.g. counterattacks) that make summaries more interesting.

The Q1 results in terms of the f-score type demonstrated the high correlation to f-score (a video with a higher f-score tends to receive a higher subjective score).

In Q2, participants gave their opinion on the summaries. Some experienced participants found the highlights too short, even the complete ones in the ground truth. This occurs because we only included in the ground truth the segments labeled as highlights by at least 40% of the participants, and thus, some labeled segments are left out. Inexperienced participants state the usefulness of our method to extract highlights based on interesting actions, and the time they can save by watching the highlights instead of the whole video. For them, incomplete highlights make the summaries hard to follow.

From this evaluation we can conclude that the labels from experienced users allow generating more interesting summaries, since they contain a better selection of techniques. Due to the negative impact of incomplete highlights on the summaries, we need to consider extra temporal consistency in p_t. We can also say that, although combining the labels of several participants is convenient to generate the ground truth, this process introduces incomplete highlights (Section 5.3.2) and alters personal preferences. We will consider instead creating personalized summaries with a higher quality ground truth, or including user profiles, such as the one proposed in [72].

5.4 Summary

In this chapter, we proposed a two-stream highlights extraction method that combines body joint-based and holistic features. The best combination among the features we evaluated are 3D body joint positions with an LSTM, and invariant spatiotemporal features (ISA) with a CNN. Users with different experience in the target sport (i.e., kendo) participated in our evaluation, where our method outperforms the previous work. Our results show that, unlike previous work, it is not necessary to recognize the players' actions explicitly to model highlights, but we can use deep learning on different representations of the players' movements. For this, LSTM has proved to be useful to model the temporal relationships of the players' joint positions and of the motion features of each video segment. In

order to successfully generate appealing summaries, features such as 3D body-joint positions and activity measure allow classifying better highlights of sports video. Generic features such as C3D and non-semantic methods such as clustering did not proved helpful for this task.

Our main contributions can be summarized as follows:

- We proposed a novel method for summarizing UGV of sports that uses two streams to extract features from the players' actions.
- We compared different feature representations of human motion and study their adequacy for modeling video highlights using a deep neural network.
- We provided an objective and subjective evaluation of our method. We surveyed users with different levels of experience in the sport to investigate the adequacy of our method to their particular preferences.

6 Conclusion

This thesis described a novel approach to user-generated sports video summarization. Sports video summarization methods so far focused on leveraging editing conventions of a specific target sport to detect the highlights of the game. These heuristic methods are not applicable to user-generated video since it is mostly unstructured and unedited. However, current UGV summarization methods are quite general and do not really approach sports video directly, which makes them inappropriate to extract sports video highlights (Chap. 2). We tackle the user-generated sports video summarization problem by relying on a source of features that is common to all UGV of sports, the players. We hypothesize that by using the players' actions as features we can generate a summary of a sports game. In a first attempt to prove our hypothesis, we proposed a method that used the results of applying human action recognition to the players' motion to model highlights using a GMM-HMM. 3D human motion representation, such as 3D joint positions estimated using depth maps or MoCap sequences, offers the most accurate recognition results. However, due to the lack of user-generated sport video datasets using this representation, we recorded our own RGB-D sports dataset, which had a limited number of actions. Faced with this problem, we designed a HAR method for 3D joint trajectories that allows recognizing actions with a limited number of training instances (Chap. 3), and used it in our summarization approach. We evaluated the method using the annotations of people with and without experience in the sport. The objective results and the positive responses from the survey showed that the players' actions can actually be used to generate summaries of UGV of sports, extracting different highlights depending on the person who annotated the training videos (experienced or inexperienced) (Chap. 4). To the best of our knowledge, this was the first time ever that user-generated sports video summarization has been approached directly in this way, making this re-

search a very novel work. One of the conclusions obtained was that we do not need to recognize the actions perfectly in order to model the highlights. Thus, we improved our original summarization method by considering not only the 3D information of our videos, but also the motion cues present in the RGB frames. We used convolutional neural networks to automatically extract those cues and, instead of performing HAR, we fed them to a recurrent neural network to model the interesting highlights (Chap. 5). The results show we surpassed our previous method and the new extracted features opened a way for future research in the UGV summarization field.

The main conclusions of this thesis are summarized as follows:

- Even in the case of not having a large action dataset, using 3D information and template matching can provide good recognition results.
- The confidence value of a 3D joint position estimator can be used to filter the noise of the capturing device.
- The player's actions can be used as semantic features to model the highlights in user-generated sports video.
- The activity measure feature rises the performance notable for highlight modeling in all the cases we evaluated.
- Our method extracts different highlights depending on the level of experience in the sport of the user who annotated the training data.
- It is not necessary to recognize the players' actions explicitly to generate a highlights summary with our method.
- We obtained the best results using a two-stream highlights extraction that combines coarse and detailed motion features. The best combination among the features we evaluated are 3D body joint positions with an LSTM, and invariant spatiotemporal features with a CNN.
- LSTM has proved to be useful to model the temporal relationships of the players' joint positions and of the motion features of each video segment.

- In order to successfully generate appealing summaries (especially to experienced users), features such as detailed motion and activity measure allow distinguishing skillful actions from poorly-executed actions. Generic features such as C3D did not proved helpful for this task.

For the future work in our user-generated sports video summarization approach, we will extend this method to a variety of sports (boxing, martial arts, etc.) and other types of videos where people are protagonists, such as dance performances, concerts, etc. We will investigate a way to include context into the extracted highlights. Another research direction is to explore different motion features and models for a better highlight extraction. We also plan to extend our dataset by using videos *in the wild*, i.e., user-generated and publicly available. In the near future next-generation devices will feature new sensors that will allow them to capture 3D information. For example, the new iPhone7 Plus has two cameras, which allows to generate depth maps for advanced image processing. This will provide many opportunities to extract a variety of motion features from user-generated video. As for the future work in our flexible HAR method, we plan to optimize the generation of action templates by eliminating redundant information (i.e. clustering similar instances or forgetting unused instances), and therefore reducing classification times. We will also address the recognition of action classes that only differ in their speed (e.g. touching and punching).

The proposed summarization method can be employed in several applications. The most direct one is extracting the highlights of lengthy user videos of sports to facilitate their review, transmission, etc. But also, the extracted features can be used for video indexing according to which actions the players are doing. It could be useful to look for certain patterns in the players' actions and obtain statistics about the game, and coaches could use this data to analyze the performance of their players, etc. Our proposed flexible HAR method is very promising when applied to customizable gesture interfaces, where a user could input, modify, and delete actions in real-time without needing to retrain the system.

Acknowledgements

I would like to express my gratitude to Professor Naokazu Yokoya, who gave me the opportunity to enter the Vision and Media Computing Laboratory and kindly signed the letter of acceptance for the MEXT scholarship. Without his support this dissertation would not have been possible. I would like to thank Associate Professor Tomokazu Sato, who took me as his student and helped me in my research. I am grateful to Affiliate Assistant Professor Yuta Nakashima for his close supervision during my PhD studies and his strict teachings. Also, I am thankful to Assistant Professor Norihiko Kawai for his constructive comments and suggestions. I am also very grateful to Professor Hirokazu Kato who was part of the committee in several of my presentations.

I would like to thank Professor Mario Martínez Zarzuela and Professor Francisco Javier Díaz Pernas from University of Valladolid, who introduced me to their research in computer vision eight years ago, and constantly offered me chances to collaborate with them. Thank you for your guidance and support. I am also grateful to Professor Janne Heikkilä, Professor Esa Rahtu, and Marko Linna from University of Oulu, who welcomed me for a short internship, and offered me their feedback and ideas in the last part of my doctoral course. I want to express my gratitude to Professors Mike Barker, David Sell, Adarsh Sharma, Toshinori Takai, and Yasushi Tanaka, who allowed me and encouraged me to participate in their classes.

I would like to thank secretaries Yumi Ishitani and Azusa Minami, who assisted me with daily paperwork, always with a smile. I am very grateful to the other PhD candidates Fabian Lorenzo Baytion Dayrit, Mayu Otani, Hikari Takehara for sharing many anecdotes and helping me in my research and the CICP projects, which I could not have won without them. I would also like to thank the other CICP teammates: Andrei Tuchin, Do Quoc Truong, Nurul Fithria Lubis,

and Muhaimin Hading. And the UX Design teammates Takuto Norikane, Aya Nakata, Kenichi Ono, Fumika Morimoto, Natsumi Saruwatari and Yudai Nakaya, who fought hard to win the competition and made me feel like a true leader. I also want to say thank you to all Master students from the laboratory, those who graduated and those who will graduate.

To my friends from NAIST, Gustavo Alfonso García Ricárdez, Felix von Drigalski, Lotfi El Hafi, Stevce Radevski, Jirayus "Yo" Jiarpakdee, Juan Esteban Rodríguez Ramírez, Ander Martínez, Pedro Urigüen Eljuri, and Konan Cedric. Thank you for making every moment we spent together fun, and sharing the complex feelings of living in Japan. Thank you, Akpa Elder, Gian Mayuga, Rodrigo Elizalde-Zapata and Yoshi Komura for performing on stage and getting a standing ovation together. Special mention to Christopher Michael Yap for his personal support and mutual understanding when looking for a direction in our careers and lives. I am deeply grateful to NAIST Kendo club members, Chaiyanan "Oat" Kulchaisit, Risaku Hirai, Nozomi Terasaki, Yuya Iwaguchi, and Kimihiko Nakatani for their efforts in creating the club and its activities. I want to thank NAIST science communicator Izumi Dateyama for giving me the chance to prepare and teach my first seminar. I would like to express my appreciation to NAIST international students affairs section and division for global education for their warm support in my university life.

I would like to offer my special thanks to the Vulcanus in Japan Programme, which allowed me to come to Japan in the first place, and my Vulcanus partners in Atsugi: Gianfranco D'Ambrosio, Jose Alberto Rodríguez Santamaría, Krzysztof Gibasiewicz, and Peter Karkus, who showed me the joy of being a foreigner in Japan. My special appreciation to Kendo teachers Kenji Takizawa, Masaya Takizawa and Shuji Amano, and Iaido teachers Teruo Ikeda, Yasushi Yamamoto and Masashi Matsumoto for welcoming me into their dojo and teaching me patiently the Japanese soul. I also want to express my gratitude to Asami Hatano, for teaching me my first Japanese language lessons for four years. And thank you sincerely Mariko Suzuki for showing me so much about myself in so little time.

And finally, my biggest and deepest thanks to my family: Antonio, Margarita, and Miguel Ángel, and Spanish friends: Castro, Adri, Valli, and Jordi, for their unconditional support from such a long distance, and because they always be-

lieved in me even when I did not. I hope someday I can give you back all the love.

References

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1–16:43, 2011.

[2] J. K. Aggarwal and L. Xia. Human activity recognition from 3D data: A review. *Pattern Recognition Letters*, 48:70–80, 2014.

[3] G. Al-Naymat, S. Chawla, and J. Taheri. SparseDTW: A novel approach to speed up dynamic time warping. In *Proc. the 8th Australasian Data Mining Conference-Volume 101*, pages 117–127, 2009.

[4] B. B. Amor, J. Su, and A. Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):1–13, 2016.

[5] G. Ballin, M. Munaro, and E. Menegatti. Human action recognition from RGB-D frames based on real-time 3D optical flow estimation. In *Proc. Biologically Inspired Cognitive Architectures*, pages 65–74. 2013.

[6] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute*, 4(510):1–126, 1998.

[7] K. K. Biswas and S. K. Basu. Gesture recognition using Microsoft Kinect. In *Proc. IEEE International Conference on Automation, Robotics and Applications*, pages 100–103, 2011.

[8] P. V. K. Borges, N. Conci, and A. Cavallaro. Video-based human behavior understanding: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(11):1993–2008, 2013.

[9] S. Calderara, R. Cucchiara, and A. Prati. Action signature: A novel holistic representation for action recognition. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 121–128, 2008.

[10] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(5):698–713, 1992.

[11] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero. A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 117(6):633–659, 2013.

[12] C. Y. Chen, J. C. Wang, J. F. Wang, and Y. H. Hu. Motion entropy feature and its applications to event-based segmentation of sports video. *EURASIP Journal on Advances in Signal Processing*, 2008:1–8, 2008.

[13] D. Y. Chen, S. W. Shih, and H. Y. M. Liao. Human action recognition using 2-D spatio-temporal templates. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 667–670, 2007.

[14] F. Chen, C. De-Vleeschouwer, and A. Cavallaro. Resource allocation for personalized video summarization. *IEEE Transactions on Multimedia*, 16(2):455–469, 2014.

[15] H. S. Chen, H. T. Chen, Y. W. Chen, and S. Y. Lee. Human action recognition using star skeleton. In *Proc. ACM International Workshop on Video Surveillance and Sensor Networks*, pages 171–178, 2006.

[16] M. Chen, S. C. Chen, M. L. Shyu, and K. Wickramaratna. Semantic event detection via multimodal data mining. *IEEE Signal Processing Magazine*, 23(2):38–46, 2006.

[17] W. H. Cheng, Y. Y. Chuang, Y. T. Lin, C. C. Hsieh, S. Y. Fang, B. Y. Chen, and J. L. Wu. Semantic analysis for automatic event recognition and segmentation of wedding ceremony videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1639–1650, 2008.

[18] J. Choi, W. J. Jeon, and S. C. Lee. Spatio-temporal pyramid matching for sports videos. In *Proc. ACM International Conference on Multimedia Information Retrieval*, pages 291–297, 2008.

[19] Carnegie Mellon University CMU. CMU graphics lab motion capture database. `http://mocap.cs.cmu.edu`, last visit: Dec. 15th, 2016.

[20] Y. Cong, J. Yuan, and J. Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Transactions on Multimedia*, 14(1):66–75, 2012.

[21] S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78, 1993.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[23] A. Tejero de Pablos, Y. Nakashima, T. Sato, and N. Yokoya. Human action recognition-based video summarization for RGB-D personal sports video. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 1–6, 2016.

[24] A. Tejero de Pablos, Y. Nakashima, N. Yokoya, F. J. Díaz-Pernas, and M. Martínez-Zarzuela. Flexible human action recognition in depth video sequences using masked joint trajectories. *EURASIP Journal on Image and Video Processing*, 2016(1):1–12, 2016.

[25] D. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *Proc. ACM International Conference on Multimedia*, pages 211–218, 1998.

[26] A. Divakaran, K. A. Peker, R. Radhakrishnan, Z. Xiong, and R. Cabasson. Video summarization using mpeg-7 motion activity and audio descriptors. In *Video Mining*, pages 91–121. 2003.

[27] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[28] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, 2003.

[29] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *Proc. Conference and Workshop on Neural Information Processing Systems*, pages 3468–3476, 2016.

[30] H. Fujiyoshi, A. J. Lipton, and T. Kanade. Real-time human motion analysis by image skeletonization. *IEICE Transactions on Information and Systems*, 87(1):113–120, 2004.

[31] J. Giles. Inside the race to hack the Kinect. *New Scientist*, 208(2789):22–23, 2010.

[32] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992.

[33] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. International Conference on Machine Learning*, pages 1764–1772, 2014.

[34] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van-Gool. Creating summaries from user videos. In *Proc. European Conference on Computer Vision*, pages 505–520, 2014.

[35] A. Hanjalic. Adaptive extraction of highlights from a sport video based on excitement modeling. *IEEE Transactions on Multimedia*, 7(6):1114–1122, 2005.

[36] M. Hasan and A. K. Roy-Chowdhury. A continuous learning framework for activity recognition using deep hybrid feature models. *IEEE Transactions on Multimedia*, 17(11):1909–1922, 2015.

[37] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[38] X. S. Hua, L. Lu, and H. J. Zhang. Optimization-based automated home video editing system. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):572–583, 2004.

[39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. ACM International Conference on Multimedia*, pages 675–678, 2014.

[40] Y. G. Jiang, Q. Dai, T. Mei, Y. Rui, and S. F. Chang. Super fast event recognition in internet videos. *IEEE Transactions on Multimedia*, 17(8):1174–1186, 2015.

[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[42] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations*, pages 1–13, 2015.

[43] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *Proc. British Machine Vision Conference*, pages 275–1, 2008.

[44] R. Laganière, R. Bacco, A. Hocevar, P. Lambert, G. Païs, and B. E. Ionescu. Video summarization from spatio-temporal features. In *Proc. ACM TRECVid Video Summarization Workshop*, pages 144–148, 2008.

[45] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3361–3368, 2011.

[46] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1346–1353, 2012.

[47] B. Li and M. I. Sezan. Event detection and summarization in sports video. In *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 132–138, 2001.

[48] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 9–14, 2010.

[49] R. W. Lienhart. Dynamic video summarization of home video. In *Proc. SPIE Electronic Imaging*, pages 378–389, 1999.

[50] M. Linna, J. Kannala, and E. Rahtu. Real-time human pose estimation from video with convolutional neural networks. *arXiv preprint arXiv:1609.07420*, pages 1–16, 2016.

[51] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003, 2009.

[52] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

[53] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[54] F. Lv and R. Nevatia. Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost. In *Proc. European Conference on Computer Vision*, pages 359–372, 2006.

[55] Y. F. Ma, X. S. Hua, L. Lu, and H.J. Zhang. A generic framework of user attention model and its application in video summarization. *IEEE Transactions on Multimedia*, 7(5):907–919, 2005.

[56] B. Mahasseni and S. Todorovic. Regularizing long short term memory with 3d human-skeleton sequences for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3054–3062, 2016.

[57] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936, 2009.

[58] J. Martens and I. Sutskever. Learning recurrent neural networks with Hessian-free optimization. In *Proc. International Conference on Machine Learning*, pages 1033–1040, 2011.

[59] M. Martínez-Zarzuela, F. J. Díaz-Pernas, A. Tejero de Pablos, D. González-Ortega, and M. Antón-Rodríguez. Action recognition system based on human body tracking with depth images. *Advances in Computer Science: an International Journal*, 3(1):115–123, 2014.

[60] E. Mendi, H. B. Clemente, and C. Bayrak. Sports video summarization based on motion analysis. *Computers & Electrical Engineering*, 39(3):790–796, 2013.

[61] H. Meng, N. Pears, and C. Bailey. A human action recognition system for embedded computer vision application. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[62] M. Mills, J. Cohen, and Y. Y. Wong. A magnifier tool for video data. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 93–98, 1992.

[63] P. Mistry, P. Maes, and L. Chang. WUW-wear Ur world: A wearable gestural interface. In *Proc. CHI Extended Abstracts on Human Factors in Computing Systems*, pages 4111–4116, 2009.

[64] T. K. Moon. The expectation-maximization algorithm. *Signal processing magazine*, 13(6):47–60, 1996.

[65] M. Müller, A. Baak, and H. P. Seidel. Efficient and robust annotation of motion capture data. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 17–26, 2009.

[66] M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 137–146, 2006.

[67] J. Nam and A. H. Tewfik. Dynamic video summarization and visualization. In *Proc. ACM International Conference on Multimedia (Part 2)*, pages 53–56, 1999.

[68] S. Nepal, U. Srinivasan, and G. Reynolds. Automatic detection of goal segments in basketball videos. In *Proc. ACM International Conference on Multimedia*, pages 261–269, 2001.

[69] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[70] C. W. Ngo, Y. F. Ma, and H. J. Zhang. Video summarization and scene detection by graph modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(2):296–305, 2005.

[71] H. Ning, W. Xu, Y. Gong, and T. Huang. Latent pose estimator for continuous action recognition. In *Proc. European Conference Computer Vision*, pages 419–433. 2008.

[72] N. Nitta, Y. Takahashi, and N. Babaguchi. Automatic personalized video abstraction for sports videos using metadata. *Multimedia Tools and Applications*, 41(1):1–25, 2009.

[73] O. Oreifej and Z. Liu. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

[74] H. Pan, P. Van-Beek, and M. I. Sezan. Detection of slow-motion replay segments in sports video for highlights generation. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1649–1652, 2001.

[75] H. Pazhoumand-Dar, C. P. Lam, and M. Masek. Joint movement similarities for robust 3D action recognition using skeletal data. *Journal of Visual Communication and Image Representation*, 30:10–21, 2015.

[76] W. T. Peng, W. T. Chu, C. H. Chang, C. N. Chou, W. J. Huang, W. Y. Chang, and Y. P. Hung. Editing by viewing: Automatic home video summarization by viewing behavior analysis. *IEEE Transactions on Multimedia*, 13(3):539–550, 2011.

[77] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Proc. IEEE International Conference on Robotics and Automation*, pages 3108–3113, 2010.

[78] Z. Prekopcsák, P. Halácsy, and C. Gáspár-Papanek. Design and development of an everyday hand gesture interface. In *Proc. ACM International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 479–480, 2008.

[79] L. Rabiner and B. H. Juang. Fundamentals of speech recognition. 1993.

[80] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian. HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition. In *Proc. European Conference on Computer Vision*, pages 742–757. 2014.

[81] M. Raptis, D. Kirovski, and H. Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–156, 2011.

[82] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[83] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for TV baseball programs. In *Proc. ACM International Conference on Multimedia*, pages 105–115, 2000.

[84] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proc. ACM international conference on Multimedia*, pages 357–360, 2007.

[85] S. Sempena, N. U. Maulidevi, and P. R. Aryan. Human action recognition using dynamic time warping. In *Proc. IEEE International Conference on Electrical Engineering and Informatics*, pages 1–5, 2011.

[86] E. Shechtman and M. Irani. Space-time behavior based correlation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 405–412, 2005.

[87] E. Shechtman and M. Irani. Space-time behavior based correlation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 405–412, 2005.

[88] Y. Sheikh, M. Sheikh, and M. Shah. Exploring the space of a human action. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 144–149, 2005.

[89] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[90] M. L. Shyu, Z. Xie, M. Chen, and S. C. Chen. Video semantic event/concept detection using a subspace-based multimedia data mining framework. *IEEE Transactions on Multimedia*, 10(2):252–259, 2008.

[91] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[92] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, pages 1–14, 2014.

[93] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5179–5187, 2015.

[94] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CRCV-TR-12-01*, pages 1–6, 2012.

[95] E. A. Suma, B. Lange, A. S. Rizzo, D. M. Krum, and M. Bolas. FAAST: The flexible action and articulated skeleton toolkit. In *Proc. IEEE Virtual Reality Conference*, pages 247–248, 2011.

[96] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[97] H. Tang, V. Kwatra, M. E. Sargin, and U. Gargi. Detecting highlights in sports videos: Cricket as a test case. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 1–6, 2011.

[98] D. Tjondronegoro, Y. P. Chen, and B. Pham. Integrating highlights for more complete sports video summarization. *IEEE MultiMedia*, 11(4):22–37, 2004.

[99] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: A next-generation open source framework for deep learning. In *Proc. Conference and Workshop on Neural Information Processing Systems*, pages 1–6, 2015.

[100] J. Tompkin, K. I. Kim, J. Kautz, and C. Theobalt. Videoscapes: Exploring sparse, unstructured video collections. *ACM Transactions on Graphics*, 31(4):68:1–68:12, 2012.

[101] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *2015 IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[102] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1):1–37, 2007.

[103] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[104] A. Veeraraghavan, A. Srivastava, A. K. Roy-Chowdhury, and R. Chellappa. Rate-invariant recognition of humans and their activities. *IEEE Transactions on Image Processing*, 18(6):1326–1339, 2009.

[105] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[106] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. British Machine Vision Conference*, pages 124–1, 2009.

[107] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2012.

[108] J. Wang and Y. Wu. Learning maximum margin temporal warping for action recognition. In *Proc. IEEE International Conference on Computer Vision*, pages 2688–2695, 2013.

[109] J. Wang, C. Xu, E. Chng, and Q. Tian. Sports highlight detection from keyword sequences using HMM. In *Proc. IEEE International Conference on Multimedia and Expo*, volume 1, pages 599–602, 2004.

[110] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. Ogunbona. Deep convolutional neural networks for action recognition using depth map sequences. *arXiv preprint arXiv:1501.04686*, pages 1–8, 2015.

[111] S. Wang, Z. Ma, Y. Yang, X. Li, C. Pang, and A. G. Hauptmann. Semi-supervised multiple feature analysis for action recognition. *IEEE Transactions on Multimedia*, 16(2):289–298, 2014.

[112] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3714–3722, 2016.

[113] L. Xia, C. C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 20–27, 2012.

[114] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang. Highlights extraction from sports video based on an audio-visual marker detection framework. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 1–4, 2005.

[115] G. Xu, Y. F. Ma, H. J. Zhang, and S. Q. Yang. An HMM-based framework for video semantic analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(11):1422–1433, 2005.

[116] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proc. IEEE International Conference on Computer Vision*, pages 4633–4641, 2015.

[117] D. Yow, B. L. Yeo, M. Yeung, and B. Liu. Analysis and presentation of soccer highlights from digital video. In *Proc. Asian Conference on Computer Vision*, volume 95, pages 499–503, 1995.

[118] K. H. Zeng, T. H. Chen, J. C. Niebles, and M. Sun. Title generation for user generated videos. In *Proc. European Conference on Computer Vision*, pages 609–625, 2016.

[119] K. Zhang, W. L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *Proc. European Conference on Computer Vision*, pages 766–782, 2016.

[120] Z. Zhang. Microsoft Kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, 2012.

[121] G. Zhu, C. Xu, Q. Huang, W. Gao, and L. Xing. Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. In *Proc. ACM International Conference on Multimedia*, pages 431–440, 2006.

[122] L. Zhu, Y. Chen, Y. Lu, C. Lin, and A. Yuille. Max margin and/or graph learning for parsing the human body. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

Publication List

Peer review journal papers

1. A. Tejero-de-Pablos, Y. Nakashima, N. Yokoya, F. J. Díaz-Pernas, and M. Martínez-Zarzuela. Flexible human action recognition in depth video sequences using masked joint trajectories. *EURASIP Journal on Image and Video Processing*, 2016(1):1–12, 2016. (Chap. 3)

2. M. Martínez-Zarzuela, F.J. Díaz-Pernas, A. Tejero-de-Pablos, D. González-Ortega, and M. Antón-Rodríguez. Action recognition system based on human body tracking with depth images. *Advances in Computer Science: an International Journal*, 3(1):115–123, 2014. (Chap. 3)

3. A. Tejero-de-Pablos, and I. de la Torre. Advances and current state of the security and privacy in Electronic Health Records: Survey from a social perspective. *Journal of Medical Systems*, 36(5):3019–3027, 2012.

Peer review international conferences

1. A. Tejero-de-Pablos, Y. Nakashima, T. Sato, and N. Yokoya. Human action recognition-based video summarization for RGB-D personal sports video. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 1–6, 2016. (Chap. 4)

2. M. Martínez-Zarzuela, F. J. Díaz-Pernas, A. Tejero-de-Pablos, F. Perozo-Rondón, M. Antón-Rodríguez, and D. González-Ortega. Fuzzy ARTMAP based neural networks on the GPU for high-performance pattern recognition. In *Proc. International Work-Conference on the Interplay Between Natural and Artificial Computation*, pages 343–352, 2011.

3. M. Martínez-Zarzuela, F. J. Díaz-Pernas, A. Tejero-de-Pablos, M. Antón-Rodríguez, J. F. Díez-Higuera, D. Boto-Giralda, and D. González-Ortega. Adaptive resonance theory fuzzy networks parallel computation using CUDA. In *Proc. International Work-Conference on Artificial Neural Networks*, pages 150–157, 2009.

Peer review domestic conferences

1. M. Martínez-Zarzuela, F. J. Díaz-Pernas, A. Tejero-de-Pablos, F. Perozo-Rondón, M. Antón-Rodríguez, and D. González-Ortega. 3D human body tracking using Kinect technology. In *Proc. Seminario Anual de Automática, Electrónica Industrial e Instrumentación*, pages 747–752, 2011 (in Spanish). (Chapter 3)

Patents

1. A. Tejero-de-Pablos, R. García-Martínez, A. Cid-Rodríguez, M. Martínez-Zarzuela, F. J. Díaz-Pernas, and F. Perozo-Rondón. "3D environments builder with virtual and augmented reality." Registration Entry Number: 00/2013/1628. Issued in Spain in January 2013.

Awards

1. Special Research Project Award, Creative and International Competitiveness Project (CICP) 2015: "Motion-based sports learning with natural language interaction", NAIST, February 2016.

2. First prize, UX Design Contest 2015: "My Camera, collaborative system for photo advices in real-time", NAIST and Osaka Arts University, December 2015.

3. Special Research Project Award, Creative and International Competitiveness Project (CICP) 2014: "Multimedia abnormal detection for elders via action and pulse recognition", NAIST, February 2015.

Appendix: Self-generated UGV sports dataset

ID	Original video	Ground truth *E*	Ground truth *NE*
#1	10 min 48 sec	1 min 11 sec	2 min 21 sec
#2	5 min 10 sec	49 sec	1 min 7 sec
#3	5 min 18 sec	1 min 9 sec	1 min 58 sec
#4	9 min 37 sec	1 min 37 sec	2 min 17 sec
#5	9 min 59 sec	2 min 33 sec	2 min 42 sec
#6	10 min 5 sec	1 min 28 sec	2 min 55 sec
#7	10 min 3 sec	48 sec	1 min 45 sec
#8	10 min 10 sec	45 sec	2 min 14 sec
#9	5 min 17 sec	32 sec	1 min 14 sec
#10	5 min 14 sec	22 sec	1 min 30 sec
#11	4 min 58 sec	53 sec	1 min 50 sec
#12	20 min 40 sec	1 min 24 sec	4 min 14 sec
#13	10 min 15 sec	53 sec	2 min 50 sec
#14	10 min 16 sec	58 sec	5 min 8 sec
#15	10 min 37 sec	47 sec	2 min 44 sec
#16	10 min 37 sec	34 sec	2 min 21 sec
#17	5 min 14 sec	16 sec	1 min 44 sec
#18	5 min 4 sec	32 sec	2 min 21 sec
#19	10 min 57 sec	38 sec	2 min 11 sec
#20	5 min 36 sec	27 sec	1 min 21 sec
#21	5 min 36 sec	33 sec	1 min 35 sec
#22	10 min 48 sec	58 sec	1 min 59 sec
#23	9 min 44 sec	1 min 11 sec	2 min 48 sec
#24	10 min 23 sec	54 sec	2 min 25 sec
#25	10 min 7 sec	28 sec	1 min 57 sec
#26	10 min 40 sec	49 sec	2 min 5 sec
#27	4 min 59 sec	33 sec	2 min 13 sec
#28	8 min 13 sec	47 sec	2 min 10 sec
Total	4 hours 6 min 11 sec	24 min 49 sec	1 hour 3 min 59 sec